

# An efficient genetic algorithm for large-scale transmit power control of dense and robust wireless networks in harsh industrial environments

Xu Gong\*, David Plets, Emmeric Tanghe, Toon De Pessemier, Luc Martens, Wout Joseph

*Department of Information Technology, Ghent University/imec, Technologiepark 15, 9052 Ghent, Belgium*

\* E-mail: xu.gong@outlook.com. Tel.: +32 9 33 14918. Fax.: +32 9 33 14899.

## Abstract

The industrial wireless local area network (IWLAN) is increasingly dense, due to not only the penetration of wireless applications to shop floors and warehouses, but also the rising need of redundancy for robust wireless coverage. Instead of simply powering on all access points (APs), it becomes an unavoidable challenge to dynamically control the transmit power of APs on a large scale, in order to minimize interference and adapt the coverage to the latest shadowing effects of dominant obstacles in an industrial indoor environment. To tackle this challenge, this paper formulates a transmit power control (TPC) model that enables both powering on/off APs and transmit power calibration of each AP that is powered on. This TPC model uses an empirical one-slope path loss model considering three-dimensional obstacle shadowing effects, to enable accurate yet simple coverage prediction. An efficient genetic algorithm (GA), named GATPC, is designed to solve this TPC model even on a large scale. To this end, it leverages repair mechanism-based population initialization, crossover and mutation, parallelism as well as dedicated speedup measures. The GATPC was experimentally validated in a small-scale IWLAN that is deployed a real industrial indoor environment. It was further numerically demonstrated and benchmarked on both small- and large-scales, regarding the effectiveness and the scalability of TPC. Moreover, sensitivity analysis was performed to reveal the produced interference and the qualification rate of GATPC according to varying target coverage percentage as well as number and placement direction of dominant obstacles.

**Keywords:** Genetic algorithms, evolutionary optimization, large-scale optimization, cyber-physical system, Internet-of-Things

## 1. Introduction

The dominant wireless local area network (WLAN) technology IEEE802.11 or WiFi is penetrating into factories to promote factories of the future (FoF) [1]. Consequently, an industrial WLAN (IWLAN) is emerging as a basic infrastructure for manufacturing operations. For instance, production cell controllers can connect to other intelligent devices such as robot arms via an IWLAN on the shop floor [2], in order to realize agile production. The other industrial operations that increasingly rely on IWLANs are illustrated as intra-factory transportation by automated guided vehicles, video monitoring, process monitoring, etc. Compared to cabled technologies for interconnection of machines or devices, wireless technologies are superior in terms of mobility, flexibility and cheap installation and maintenance, although wireless technologies are currently less robust in harsh industrial environments. Compared to other wireless technologies, an IWLAN has the advantages of low cost, high data rate and considerable coverage distance.

However, a typical industrial indoor environment is harsh in terms of radio propagation. Firstly, either a shop floor or a warehouse is dominated by various metal facilities, such as production machines/lines, storage racks, steel bars, metal plates, pipes, automated guided vehicles, cranes and forklifts. These obstacles easily shadow radio propagation and cause coverage hole for a WLAN [1, 3]. Secondly, an industrial indoor layout may occasionally be altered with the prevalence of flexible manufacturing [4]. These dynamic shadowing effects make it increasingly difficult to maintain the expected wireless coverage in a target industrial environment. Thirdly, an IWLAN is denser compared to a public WLAN. This is not only due to the large size of an industrial indoor environment, but also driven by the increasing industrial need for redundant coverage to ensure high network availability [5]. Therefore, it is of strategic importance to conceive a transmit power control (TPC) method to dynamically change the coverage of a dense IWLAN according to these shadowing effects, in order to guarantee robust wireless connection of personnel, machines, materials and products on a large scale.

Cell breathing by TPC is a well-known concept in cellular networks [6, 7]. For instance, authors in [7] investigated a problem of minimizing total WCDMA pilot power subject to a coverage constraint. A WCDMA cell shrinks or expands according to the varying coverage rate, following the trade-off between power consumption and coverage. Comparatively, TPC of WLANs can only be found in a limited number of studies, although dense WLANs are showing up their application significance [8, 9]. A concept of resource on demand was proposed in [10] and demonstrated in [11], where redundant APs are powered off when they are detected to remain idle according to the volume and location of user demand. However, the idea of TPC beyond simple powering-on/off was only highlighted and not investigated in these studies [10, 11].

Furthermore, empirical radio propagation or path loss model is seldom used in the coverage calculation, which is essential for TPC. Power management algorithms were proposed in [12] to control the coverage of

access points (APs). However, without using any path loss model, the authors assumed that the received power of a client is proportional to the transmission power of the connected AP. Analogously, a lack of proper path loss modeling is observed in [13]. While a TPC scheme was proposed, only a linear approximation was assumed between the AP transmission power and received signal strength of a client.

The classical Boolean disk model is widely used to calculate coverage in WSN coverage related optimization problems [14, 15]. It is simple, only considering a circular area within which all grid points are covered. But its application to the IWALN coverage related optimization problems could drastically simplify the problem and degrade the optimal solution's quality, since it ignores the obstacle shadowing effects and cannot calculate the exact received RF power of a GP in the target environment. This RF power is further linked to interference estimation, which is an indispensable concern for dense WLANs [16]. On the other hand, it is costly and time consuming to undertake a complete site survey, in order to capture the actual coverage and interference. As highlighted in [17], a combination of site survey and planning algorithm design is a good method to reduce the required measurements without compromising much the coverage prediction.

While large-scale optimization is increasingly desired [18, 19], most research on coverage optimization problems neglects the scalability of an optimization algorithm [14, 20-23]. Large-scale problems are characterized in at least one of the following dimensions [24]. Firstly, the search space exponentially grows with the increasing number of decision variables. Secondly, the properties of the search space may change as the number of dimensions rises. Thirdly, the fitness evaluation is expensive. Fourthly, strong interaction exists between variables. Concerning a TPC problem, the factors that may make it large scale are illustrated as (1) the environment size, (2) the number of APs, (3) the number of coverage layers (or  $k$ -coverage), (4) the complexity of coverage calculation which is fundamentally based on a path loss model. To solve large-scale optimization problems, metaheuristics are extensively recognized as effective approaches [25], among which a genetic algorithm (GA) is an important method [26, 27].

This paper investigates a large-scale TPC problem for dense IWLANs. The contributions of this paper are threefold. (1) The proposed TPC model encompasses both transmit power calibration and powering-on/off mechanisms. An empirical one-slope path loss model is introduced for precise yet simple coverage calculation, including the three-dimensional (3D) obstacle loss which is prevalent in harsh industrial indoor environments. (2) An efficient GA, named GATPC, is proposed to solve this TPC model on a large scale. It leverages repair mechanism-based GA operators (including population initialization, crossover, and mutation), parallelism as well as dedicated speedup measures to achieve large-scale optimization. (3) The GATPC is both empirically validated in a small-scale real industrial indoor environment and extensively proved in numerical experiments regarding demonstration of effectiveness and scalability, sensitivity analysis, and benchmarking.

**Table 1**  
Nomenclature of the transmit power control model

Symbol	Meaning	Symbol	Meaning
$A$	Set of over-dimensioned access points	$\delta p$	Transmit power calibration step (dB) of an access point
$d_{ij}$	Distance between the $i$ -th grid point and the $j$ -th access point	$\vec{p}$	Vector of transmit power levels of an access point
$d_{j\max}$	Maximal radius distance the $j$ -th access point can cover with its current transmit power $P_j$	$P$	Set of transmit power levels
$G$	Total gain of a pair of transmitter and receiver	$P_j$	Transmit power level of the $j$ -th access point
$gs$	Basic grid size to discretize an environment	$P_j$	Transmit power (dBm) of the $j$ -th access point
$gp_i$	The $i$ -th grid point in an environment	$P_{ij}$	Stable power received by the $i$ -th grid point from the $j$ -th access point at 99% of the time
$I$	Set of indices of grid points	$PL0$	Path loss (dB) at the location of 1 m away from a target access point
$I_{ij}$	Interference (dBm) of the $i$ -th grid point that connects to the $j$ -th access point	$PL(d_{ij})$	Path loss (dB) between $i$ -th grid point and $j$ -th access point
$I_{ij\max}$	Maximal interference (dBm) of the $i$ -th grid point that connects to the $j$ -th access point	$Rx$	Wireless signal receiver
$J$	Set of indices of access points	$Tx$	Wireless signal transmitter
$J_{on}$	Access points that are powered on	$THLD$	Threshold received power (dBm) of a client receiver
$J_{off}$	Access points that are powered off	$xMin$	Minimal horizontal coordinate of an environment
$M$	Margin (dB) considering shadowing, fading, and interference	$xMax$	Maximal horizontal coordinate of an environment
$n$	Path loss exponent	$yMin$	Minimal vertical coordinate of an environment
$N_o$	Total number of dominant obstacles in an environment	$yMax$	Maximal vertical coordinate of an environment
$N_p$	Total number of transmit power levels (excluding powering off)	$\xi$	Deviation between measurement and model
$OL_{ij}$	Total obstacle loss (dB) between $i$ -th grid point and $j$ -th access point	$\Omega$	Set of grid points in an environment
$OL_k$	Obstacle loss (dB) of the $k$ -th dominant obstacle in the environment	$\mu$	Percentage of grid points that must be covered by at least one access point
$P$	Set of transmit power (dBm) of an access point	$\alpha_{ij}$	Local coverage variable for the $i$ -th grid point and the $j$ -th access point
$P_{\max}$	Maximal transmit power	$\beta_{ij}^k$	Logical signal blockage variable for the $i$ -th grid point, $j$ -th access point, and $k$ -th dominant obstacle
$P_{\min}$	Minimal transmit power	$\gamma_{ij}$	Logical variable for connection between the $i$ -th grid point and the $j$ -th access point

The rest of this paper is organized as follows. Sect. 2 mathematically formulates this TPC problem. Sect. 3 proposes the GATPC algorithm to solve this TPC model. Sect. 4 validates the GATPC in a small empty industrial environment. Sect. 5 performs numerical experiments, benchmarking, and sensitivity analysis of GATPC. Sect. 6 draws conclusions.

## 2. Modeling of transmit power control problem

The problem under investigation is optimal TPC of a dense WLAN in a metal-dominating industrial indoor

environment. This IWLAN is over-dimensioned such that redundant APs are planned to create double full coverage for staying robust against shadowing effects of dominant obstacles. As a result, it is unnecessary for all APs to always work at the maximal transmit power level, which produces heavy interference. Therefore, potential remains to minimize each AP's transmit power, including powering off.

A solution to this problem is denoted by  $\vec{p}$ . It is a vector of the transmit power levels of all over-dimensioned APs (denoted as  $A$ ), including the decision of powering off certain APs. Table 1 lists the nomenclature for this problem description. Sects. 2.1-2.5 will present the model of this problem in four aspects: environment modeling, transmit power setting of APs, path loss calculation, interference calculation, and objective function. Sect. 2.6 gives a brief example for this model.

### 2.1 Environment

A target rectangular environment is two-dimensional (2D), i.e., horizontal and vertical. It is represented by its two extreme 2D points (Fig. 1):  $(xMin, yMin)$  and  $(xMax, yMax)$ . It is discretized into  $gs \times gs$  small grids (Fig. 1), where  $gs$  is the grid size that is preset as an input of the model. A grid point (GP) is represented by the upper-left vertex of a grid, and denoted as  $gp_i$ , where  $i$  is a unique index for each GP. A lexicographical order is applied to all the GPs:

$$(x0, y0) < (x1, y1) \Leftrightarrow x0 < x1 \vee (x0 = x1 \wedge y0 < y1) \quad (1)$$

where  $(x0, y0)$  and  $(x1, y1)$  are illustrated coordinates for two arbitrary different GPs.

Consequently, a target environment is described by a set of ordered GPs denoted as  $\Omega$ . The GP index  $i$  within  $\Omega$  starts from one, corresponding to the extreme point  $(xMin, yMin)$  of this environment. It increases one by one until reaching  $|\Omega|$  following the lexicographical order. Then the set of GPs is denoted by their index  $I = \{1, 2, \dots, |\Omega|\}$ . The following formula determines the size of  $\Omega$ :

$$|\Omega| = \text{ceil}[(xMax - xMin) / gs] \times \text{ceil}[(yMax - yMin) / gs] \quad (2)$$

A receiver (Rx) is placed on each GP except the ones where APs are placed. The received power in the downlink is considered to enable the calculation of an AP's coverage. For an Rx, different physical bitrate requirements have different requirements on the lowest received power, named threshold ( $THLD$ ). The quantified relation can be found in [1].

The  $i$ -th GP is considered covered by the  $j$ -th AP, if an Rx on this GP connects to this AP and receives power values that are higher than or equal to the threshold during at least 99% of the time. This is formulated as follows:

$$\alpha_{ij} = \begin{cases} 1, & \text{if } P_{ij} \geq THLD \\ 0, & \text{otherwise} \end{cases}, \forall i \in I, \forall j \in J \quad (3)$$

where  $\alpha_{ij}$  is the logical coverage variable for the  $i$ -th GP and  $j$ -th AP, and  $P_{ij}$  is the stable power (dBm) that an Rx on the  $i$ -th GP receives from the  $j$ -th AP at least 99% of the time. The coverage of an AP is represented by the GPs that are covered by this AP.

## 2.2 Over-dimensioned access points

In total,  $|J|$  APs are over-dimensioned with a minimal separation distance in the environment, where  $J$  is the set of AP index which varies from one to the total number of APs ( $|A|$  or  $|\bar{p}|$ ), i.e.,  $J = \{1, 2, \dots, |A|\}$ .

In a TPC solution  $\bar{p}$ , the APs are regrouped into a set of APs that are powered off ( $J_{off}$ ) and a set of APs that are powered on with a certain power value ( $J_{on}$ ):

$$J = J_{on} \cup J_{off} \quad (4)$$

where the AP indices in  $J_{on}$  and  $J_{off}$  are still these in  $J$ .

All APs have the same TPC range  $P$  (in dBm) and step  $\delta p$  (in dB), i.e.,  $P = \{P_{\min}, P_{\min} + \delta p, P_{\min} + 2\delta p, \dots, P_{\max}\}$ . In total, there are  $N_p$  different transmit power values in  $P$ , except the possibility of powering off. Therefore,  $P_j \in P$  if  $j \in J_{on}$ , where  $P_j$  is the transmit power of the  $j$ -th AP.  $P_j$  is not considered if the  $j$ -th AP is powered off (Table 2).

**Table 2**

Mapping between physical power, power state, and the digital transmit power level of an access point

Transmit power set $P$ (dBm)	Power state	Transmit power level set $P$
-	Off	0
$P_{\min}$	On	1
$P_{\min} + \delta p$	On	2
$P_{\min} + 2\delta p$	On	3
...	On	...
$P_{\max}$	On	$N_p$

As indicated in Table 2, with the possibility of powering off, the TPC range  $P$  is discretized into  $P$ , which is a dimensionless set of all possible transmit power levels, i.e.,  $\{0, 1, 2, \dots, N_p\}$ . The discretized transmit power level of the  $j$ -th AP is denoted as  $P_j (\in P)$ . Specifically, the level zero stands for powering off. The level one represents the minimal transmit power level if an AP is powered on, and so on, until the maximal

transmit power level  $N_p$  if an AP is powered on. Consequently, the digital variable  $P_j (\forall j \in J)$  can be used to represent all the possible power states of an AP: powering off/on and if powering on, at which transmit power this AP works.

### 2.3 Path loss

In [7], the path loss model was simplified as a multiplier of transmit power. In [14, 15], the Boolean disk-based path loss model only determined whether an Rx was within the circular coverage of a Tx. In the proposed TPC model, a one-slope path loss model considering metal obstacle shadowing loss along the propagation path is considered for accurate path loss calculation, which is the basis for calculating the received power of a Rx and interference. In total, there are  $N_o (\geq 0)$  dominant obstacles in the investigated environment. This path loss model is formulated as:

$$PL(d_{ij}) = PL0 + 10n \log_{10}(d_{ij}) + OL_{ij} + \xi \quad (5)$$

where  $PL0$  (in dB) is the path loss at the distance of one meter,  $n$  is the path loss exponent which is a dimensionless parameter indicating the increase of path loss with the distance,  $d_{ij}$  is the distance (in m) between the Rx placed on the  $i$ -th GP and the  $j$ -th AP,  $OL_{ij}$  is the total obstacle loss (in dB) caused by the metal obstacles that block the line between the Rx placed on the  $i$ -th GP and the  $j$ -th AP, and  $\xi$  (in dB) is the deviation between the measurement and the model, which is attributable to shadowing.

For an investigated environment, it assumes that the obstacle locations are fixed. The deviation  $\xi$  in Eq. (5) follows a Gaussian distribution, with a mean of zero and a standard deviation  $\sigma$ . The gain and margin are considered in the link budget calculation to be more realistic, which was not taken into account in [7]. The total gain  $G$  (in dB) is the sum of the AP transmitter's gain and the Rx's gain. The margin  $M$  (in dB) is the sum of shadowing, fading and interference margin.

The total obstacle loss between the Rx on the  $i$ -th GP and the  $j$ -th AP is calculated in the following two equations. Eq. (6) iterates all the dominant obstacles in the environment and accumulates the additional path loss caused by the obstacles that blocks in the line-of-sight radio propagation from the  $j$ -th AP to the Rx on the  $i$ -th GP. Eq. (7) defines the logical signal blockage variable  $\beta_{ij}^k$ . If the  $k$ -th dominant obstacle has the shadowing effect on the line-of-sight radio propagation from the  $j$ -th AP to the Rx on the  $i$ -th GP, it equals one. Otherwise, it equals zero. The calculations defined by Eqs. (6, 7) are only limited to APs that are powered on.

$$OL_{ij} = \sum_{k=1}^{N_o} \beta_{ij}^k \cdot OL_k, \forall i \in I, \forall j \in J_{on}, \forall k \in \{1, \dots, N_o\} \quad (6)$$

$$\beta_{ij}^k = \begin{cases} 1, & \text{if the } k\text{-th metal blocks the line between} \\ & \text{the } i\text{-th GP and the } j\text{-th AP} \\ 0, & \text{otherwise} \end{cases}, \forall i \in I, \forall j \in J_{on}, \forall k \in \{1, \dots, N_o\} \quad (7)$$

Furthermore, compared to most coverage-related optimization problems that only rely on a 2D environment [1, 14, 15, 17, 20, 21], an obstacle is modeled as a 3D geometrical model in the decision making of line-of-sight propagation between a GP-AP pair (i.e., the logical signal blockage variable  $\beta_{ij}^k$ ). Both the  $j$ -th AP and the Rx placed on the  $i$ -th GP have their own heights. An obstacle has a 3D dimension of length  $\times$  width  $\times$  height. An obstacle blocks the line-of-sight propagation as long as part of it crosses the straight line between the top of the  $j$ -th AP and the top of the Rx on the  $i$ -th GP. A detailed discussion on the 3D obstacle loss calculation can be found in [28].

If the  $j$ -th AP is powered on with the transmit power  $P_j$ , the maximal radius distance this AP can cover ( $d_{j\max}$ ) can then be calculated, without considering the additional shadowing effects that may be caused by dominant obstacles. For an AP that is powered off,  $d_{j\max}$  is zero, indicating that it cannot cover any GP. This is formulated as follows:

$$d_{j\max} = \begin{cases} 10^{\left(\frac{P_j + G - M - THD - PL0}{10n}\right)}, & \forall i \in I, \forall j \in J_{on}, P_j \in P \\ 0, & \forall j \in J_{off} \end{cases} \quad (8)$$

## 2.4 Interference

An inevitable goal of transmit power management for a dense WLAN is the interference among APs. While dedicated frequency planning is out of scope in this paper, it is assumed that non-overlapping channels are effectively allocated to the dense APs. If an Rx on the  $i$ -th GP connects to the  $j$ -th AP ( $j \in J_{on}$ ), the interference ( $I_{ij}$ , in dBm) to this Rx is then all the power this Rx can sense from the other APs that are powered on ( $\forall j' \in J_{on}, j' \neq j$ ) [17, 29]. The interference calculated this way is also interpreted as noise [17]. If an AP is powered off, it is not considered by this calculation. This is formulated in the following two equations:

$$I_{ij} = 10 \cdot \log_{10} \sum_{j' \in J_{on}} 10^{\frac{P_{j'}}{10}}, \forall i \in I, \forall j, j' \in J_{on}, j' \neq j, P_{j'} \in P \quad (9)$$

$$P_{ij} = P_j + G - M - PL(d_{ij}), \forall i \in I, \forall j \in J_{on} \quad (10)$$



The worst case is that all APs are powered on with the maximal transmit power ( $P_j = P_{\max}, \forall j \in J_{on}, J_{on} = J$ ). Then the maximal interference ( $I_{ij\max}$ , in dBm) to an Rx can be calculated as follows:

$$I_{ij\max} = 10 \cdot \log_{10} \sum_{j' \in J_{on}} 10^{P_{ij'}/10}, \forall i \in I, \forall j, j' \in J_{on} = J, j' \neq j, P_j = P_{\max} \quad (11)$$

## 2.5 Transmit power control

The TPC model is minimization of normalized total interference (Sect. 2.4) under the constraint of wireless coverage (Sect. 2.3) of a metal-dominating industrial environment (Sect. 2.1) which is deployed with over-dimensioned APs (Sect. 2.2). It is described in the following three formulae.

$$\text{Objective: } \min_{P_j} \left( \frac{\sum_{i=1}^{N_{GP}} \sum_{j=1}^{|A|} 10^{I_{ij} \cdot \gamma_{ij}/10}}{\sum_{i=1}^{N_{GP}} \sum_{j=1}^{|A|} 10^{I_{ij\max} \cdot \gamma_{ij}/10}} \cdot 100\% \right), \forall p_j \in p, \forall i \in I, \forall j \in J \quad (12)$$

s. t.:

$$\sum_{j=1}^{|A|} \alpha_{ij} \geq 1, \forall i \in \mu \cdot I \quad (13)$$

$$\gamma_{ij} = \begin{cases} 1, & \text{if Rx on the } i\text{-th GP connects to the } j\text{-th AP} \\ 0, & \text{otherwise} \end{cases}, \forall i \in I, \forall j \in J \quad (14)$$

Eq. (12) sets the object of TPC as minimizing the normalized interference (in mW) in the whole over-dimensioned network. The essential variable that is tunable for this optimization is the transmit power level of each AP ( $P_j, \forall j \in J$ ) deployed in the environment.

Eq. (13) sets the constraint that a percentage  $\mu$  of all the GPs must be covered by at least one AP, i.e., a coverage rate  $\mu$  ( $\mu \in (0, 1]$ ) must be ensured in the target environment.

A logical variable of AP connection  $\beta_{ij}$  is introduced in Eq. (14). If an Rx can sense multiple APs that are powered on, it connects to the one that achieves the highest received power at this Rx. If there are multiple APs that have the same highest received power at this Rx, the Rx randomly connects to one of these APs. An Rx can connect to at most one AP, while an AP can have multiple Rx that connect to it. While received power of a client plays a vital role in handover and AP association, further discussion on client-AP association mechanism is out of scope in this paper.

Overall, the entire TPC model is mathematically formulated by Eqs. (1-14), and named the interference minimization based TPC model (IM-TPC). According to the definition of large-scale problems [24], the scale of a TPC model is influenced by (1) the size of a target industrial indoor environment (which is linked to the

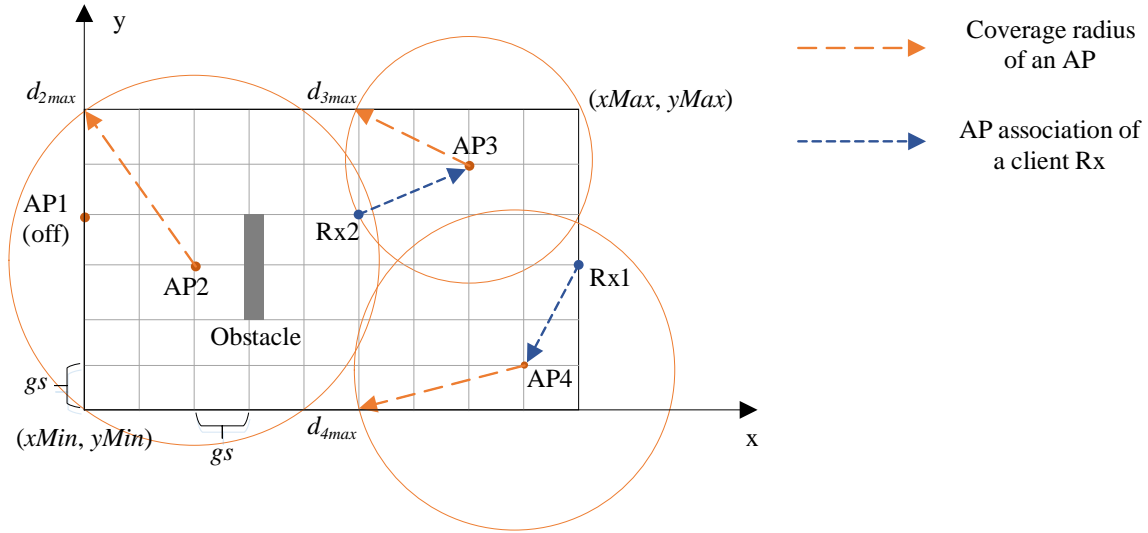


Fig. 1. Illustration of the transmit power calibration model: discretized environment with the presence of a dominant obstacle, transmit power minimization of over-dimensioned access points (APs) for only one full coverage layer, AP association of two client receivers (Rx), and interference (produced by all APs that are powered on and not connected to).

number of APs), (2) the grid size ( $gs$ ), (3) the number of dominant obstacles, and (4)  $\mu$ . The first factor determines the dimensionality of a search space and impacts the complexity of fitness evaluation. The second factor is associated to the density of a search space. The third factor defines the expense of path loss calculation and thereby fitness evaluation. The last factor controls the hardness of coverage constraint and also the interaction between transmit power levels of APs due to the planned redundancy. Therefore, a TPC model is considered as large-scale if the industrial indoor environment is large with a small  $gs$ , the presence of multiple dominant obstacles, and a high rate of  $\mu$ .

## 2.6 Illustrative example

The TPC model is illustrated in a simple example (Fig. 1). A rectangular environment, defined by  $(xMin, yMin)$  and  $(xMax, yMax)$ , is discretized into  $9 \times 6 = 54$  grids, each of which is represented by its upper-left vertex (GP). A  $gs \times gs$  grid is considered covered by an AP, if its GP is within the coverage radius of this AP which works at its current transmit power. Although 4 APs are deployed as redundancy for robustness, AP1 is powered off and the other APs work below their maximal transmit power, in order to reduce interference while guaranteeing one full coverage layer ( $\mu = 1$ ). Therefore, the 3 APs that are powered on have their respective coverage radius:  $d_{2max}$ ,  $d_{3max}$ , and  $d_{4max}$  (without considering any obstacles). There are two clients: Rx1 and Rx2. While Rx1 connects to AP4 as it is only covered by AP4, Rx2 has to decide between AP2 and AP3. Due to the obstacle shadowing between AP2 and Rx2, Rx2 receives lower signal strength from AP2 than from AP3. Rx2 thus connects to AP3. As a result of these AP associations, AP2 and AP3 produce interference to Rx1, while AP2 and AP4 cause interference to Rx2. As there are multiple transmit power

calibration solutions for one full coverage layer, the optimal solution has to be determined in terms of minimal interference. An optimization algorithm is needed for automatic decision making of the transmit power of each AP, when the TPC problem size increases and power calibration is frequently performed due to dynamic shadowing effects in harsh industrial indoor environments.

### 3. Solution algorithm

As an optimum rectangular grid coverage problem is NP-complete [30], metaheuristics are an effective technique to solve this type of problem. As a well-known metaheuristic, a GA gives a near-optimum solution for a NP-complete combinatorial problem within a reasonable time [31, 32]. This characteristic complies with the objective of the investigated TPC problem to fast obtain a high-quality solution without necessarily requiring the real optimum. Therefore, a GA is used for this TPC problem.

The design of this GA follows the objective of simultaneously minimizing memory usage and CPU time for solving a large-scale TPC model. This GA based TPC algorithm is named GATPC. Given that the GATPC checks the coverage of a GP by iterating all APs that can potentially cover this GP, the following definitions are made to facilitate the presentation of GATPC in the subsections.

**Definition 1:** *covered GPs* refer to a set of GPs that are covered by at least one AP.

**Definition 2:** *uncovered GPs* represent a set of GPs that are not yet covered by any AP with the current TPC solution  $\vec{p}$ .

**Definition 3:** *new covered GPs* of a given AP stand for a subset of *uncovered GPs* that can be covered by this AP at its current transmit power level.

**Definition 4:** a *GP-AP link* shows that the investigated GP can be covered by the investigated AP at its maximal transmit power level. It thus shows an AP's potential to cover a GP.

**Definition 5:** the *nearest potential AP* of an uncovered GP is the AP that is the nearest to this GP among all the APs that have *GP-AP links* with this GP.

#### 3.1 Solution encoding and fitness evaluation

As introduced in Sect. 2, a TPC solution is  $\vec{p}$ , a vector containing  $|\vec{p}|$  discretized AP transmit power levels, including powering off (Table 2). The index of a value in  $\vec{p}$  corresponds to the index of the AP that is over-dimensioned in the environment. The list of over-dimensioned APs is sorted by applying the lexicographical order (Eq. (1)) to the GPs on which these APs are placed. Therefore, the  $j$ -th value in  $\vec{p}$  corresponds to the transmit power level of the  $j$ -th AP. Fig. 2 illustrates an example of encoding and decoding  $\vec{p}$  in this manner.

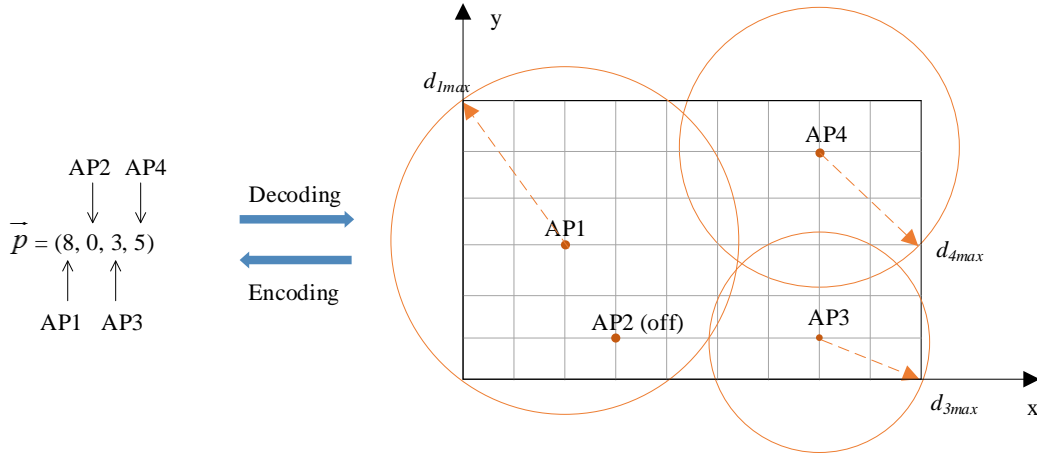


Fig. 2. Example of transmit power calibration solution encoding and decoding. A solution is denoted by a vector  $\vec{p}$  that indicates an access point (AP) and its transmit power level by its element index and element value, respectively. As a result of this  $\vec{p}$ , the relationship among coverage radiuses of all APs are:  $d_{1max} > d_{4max} > d_{3max}$ , while AP2 is powered off.

---

**Algorithm 1** Generation of a random TPC solution (RTPC)

---

**Input:** none

**Output:** a qualified random TPC solution  $\vec{p}$  satisfying the coverage constraint defined by Eq. (13)

1.  $\vec{p} \leftarrow |A|$  random numbers  $\in P$ ;
  2.  $\vec{p} \leftarrow \text{repair}(\vec{p})$ ;
- 

This encoding scheme has two advantages. The first is to facilitate the definition of crossover and mutation operations, which will be described in Sect. 3.3 and Sect. 3.4, respectively. The second advantage is efficient memory utilization. As each scalar in  $\vec{p}$  can be represented only by a 32-bit integer, little encoding memory is needed even for evolutionary optimization of a large-scale TPC problem. For instance, 1000 TPC solutions for 100 APs only occupy 0.38 MB.

The fitness of a TPC solution is normalized interference defined by Eq. (12). The entire TPC model (Sect. 2) is needed for this calculation. Therefore, the evolution of population in a GA will autonomously reduce the interference, which is the objective of TPC.

### 3.2 Population initialization

It is not obliged to generate all qualified initial individuals, since unqualified individuals will be either eliminated by the population evolution or improved by the crossover and mutation operations. However, any generation of unqualified individuals will produce computation redundancy to the GA search and thus reduce the optimization efficiency. Especially for evolutionary optimization of a large-scale TPC problem, the computation time to get an acceptable solution is quite sensitive to computation redundancy. Hence, the proposed initial population generation algorithm aims to produce 100% qualified initial individuals.

---

**Algorithm 2** Repair of an unqualified TPC solution
 

---

**Input:** a TPC solution  $\vec{p}$  without knowing its qualification

**Output:** a qualified TPC solution  $\vec{p}$  satisfying the coverage constraint defined by Eq. (13)

```

1. covered GPs  $\leftarrow$  new covered GPs of  $APs \in A$  with  $\vec{p}$ ;
2. if ( $|covered\ GPs| < \mu \cdot \Omega$ )
3.   uncovered GPs  $\leftarrow \Omega$ ;
4.   remove covered GPs from uncovered GPs;
5.   for  $j \leftarrow 1: |A|$ 
6.     if ( $\vec{p}(j) < N_p$ )
7.       set up GP-AP links between uncovered GPs and  $AP(j) \in A$ ;
8.     end if
9.   end for
10.  while (uncovered GPs  $\neq \emptyset$  &&  $|covered\ GPs| < \mu \cdot \Omega$ )
11.    find nearest potential AP of a random  $GP \in uncovered\ GPs$ ;
12.    if (nearest potential AP  $== \emptyset$ )
13.      remove this random GP from uncovered GPs;
14.    else
15.      assign nearest potential AP with the minimal transmit power level that can cover this random GP;
16.      remove new covered GPs of nearest potential AP from uncovered GPs;
17.      covered GPs  $\leftarrow new\ covered\ GPs \cup covered\ GPs$ 
18.      if (transmit power level of nearest potential AP  $== N_p$ )
19.        remove GP-AP links between nearest potential AP and all the related GPs in uncovered GPs;
20.      end if
21.    end if
22.  end while
22. end if

```

---

Algorithm 1 describes a two-step sequential method to randomly produce a qualified individual. Step 1 (line 1) randomly generates a TPC solution without considering the coverage constraint defined by Eq. (13). Step 2 (lines 2) checks and ensures the random solution's satisfaction of coverage constraint by potentially raising the transmit power of selected APs to the minimal degree. To this end, a procedure for qualification check and potential repair is proposed in Algorithm 2.

At the start of Algorithm 2, the GPs' coverage information is updated by setting APs with these input transmit power levels (line 1). If the required coverage rate is not yet achieved (line 2), this random TPC solution will be corrected (lines 3-22). The idea of repair is to iteratively cover the *uncovered GPs* using the potential APs (whose current transmit power is below the maximal level) while maintaining the increase in transmit power as slight as possible (to comply with interference minimization). At the first step of correction (lines 3-9), *uncovered GPs* are all found and their *GP-AP links* are all established. The second step of correction (lines 10-22) is an iteration. In each iteration, a GP is randomly selected from *uncovered GPs* and its *nearest potential AP* is found (line 11). The rationale behind the random selection is to increase the diversity of population and prevent premature convergence. In a minor case where a randomly selected GP is shadowed by an obstacle such that it cannot be covered by any AP, this GP is removed from *uncovered*

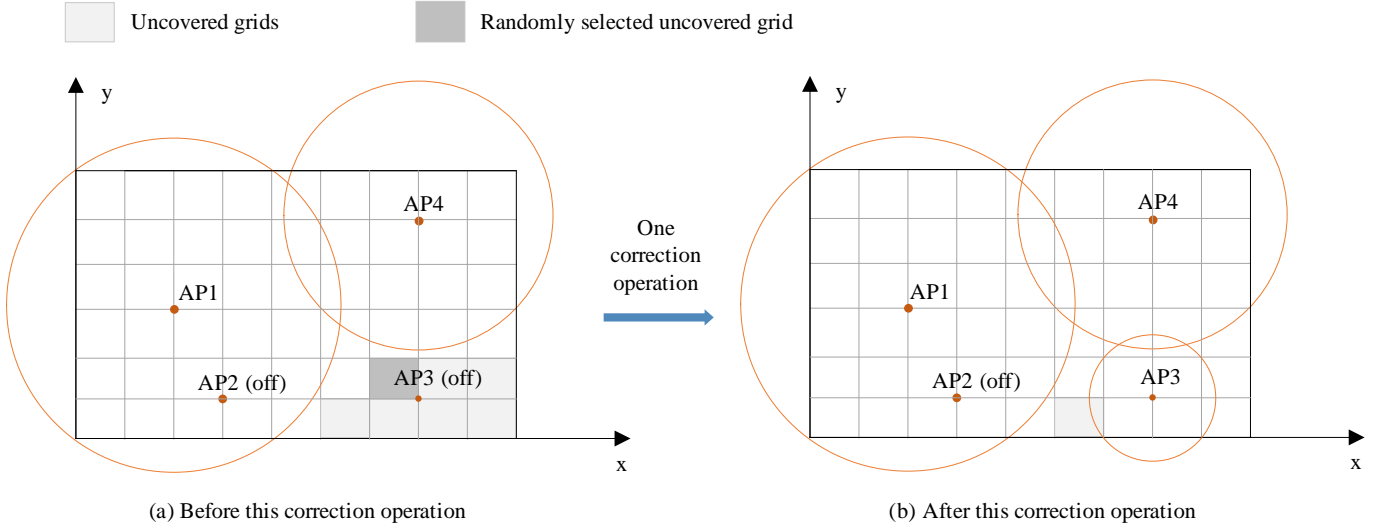


Fig. 3. Example of iterative correction of an unqualified transmit power calibration solution. (a) 7 grids are uncovered by any APs. One grid is randomly selected (dark grey) and then AP3 is selected from the 4 APs as it is the closest to this random uncovered grid (a grid is represented by its upper-left vertex). (b) 1 grid remains uncovered after AP3 is set by the minimal transmit power that can cover this random uncovered grid. Another iterative correction is thereby needed to fully cover the entire environment.

---

**Algorithm 3** Crossover for GATPC

---

**Input:** two selected parent TPC solutions *parent1* and *parent2*

**Output:** two offspring TPC solutions *offspring1* and *offspring2*

1.  $loci \leftarrow$  a randomly selected index of *parent1*;
  2. divide *parent1* and *parent2* into two parts along this random *loci*, respectively
  3.  $offspring1 \leftarrow$  1<sup>st</sup> part of *parent1* + 2<sup>nd</sup> part of *parent2*;
  4.  $offspring2 \leftarrow$  1<sup>st</sup> part of *parent2* + 2<sup>nd</sup> part of *parent1*;
  5. **for**  $indiv \in \{offspring1, offspring2\}$
  6.      $indiv \leftarrow repair(indiv)$  using Algorithm 2
  7. **end for**
- 

GPs (lines 12-13). Otherwise, the transmit power of *nearest potential AP* is set to the minimal level that can cover this random *uncovered GP* (line 15), followed by updating GP coverage information and *GP-AP links* (lines 16-20). Fig. 3 further presents an example to iteratively correct an unqualified solution (lines 3-22, Algorithm 2).

To produce the entire initial population, Algorithm 1 is iterated for a number of times equal to the size of population. As it is extremely hard to produce even a qualified solution for this TPC problem, Algorithm 1 also serves as a random TPC solution generation algorithm (named RTPC) for benchmarking.

### 3.3 Selection and crossover

Parent solutions are selected using roulette wheel selection for a crossover operation. A crossover operation enables two parent solutions to breed two new child solutions by swapping the parents' genes. A one-point crossover is realized in GATPC using two steps. Step 1 (lines 1-4, Algorithm 3) exchanges partial

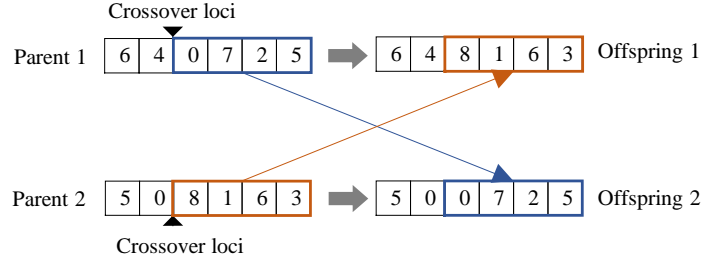


Fig. 4. Example of one-point crossover of two parent transmit power calibration solutions, which exchange a segment of transmit power levels in vector  $\vec{p}$ . The offspring would need to be corrected.

---

**Algorithm 4** Mutation for GATPC

---

**Input:** an offspring TPC solution output by crossover and selected for mutation

**Output:** a new TPC solution

1.  $selectedAPs \leftarrow AP(s)$  in the input individual that reach(es) the highest transmit power level ( $N_p$ );
  2. **if** ( $|selectedAPs| > 1$ )
  3.      $selectedAP \leftarrow$  a random AP in  $selectedAPs$ ;
  4. **end if**
  5. new  $\vec{p} \leftarrow$  power off  $selectedAP$ ;
  6.  $\vec{p} \leftarrow repair(\vec{p})$  using Algorithm 2
- 

chromosomes of two parent solutions around a randomly selected loci. Fig.4 illustrates this one-point crossover operation. It is simple and intuitive due to the encoding scheme (Sect. 3.2) and lexicographical ordering of all APs in Eq. (1). Step 3 (lines 5-7 in Algorithm 3) first checks whether each child solution achieves full coverage. If the coverage rate, required by Eq. (13), is not yet achieved, the *uncovered GPs* will be addressed one by one with their *nearest potential APs*. This potential correction of an unqualified solution follows the repair procedure described in Algorithm 2.

### 3.4 Mutation

A GA is known as a global optimization algorithm. A mutation operation plays a vital role to this end. The mutation of GATPC is defined by Algorithm 4, comprising two steps. Step 1 (lines 1-5) powers off one AP that already reaches the highest transmit power level, i.e., without potential to increase transmit power any more. This aims to increase the diversity in the solution space and presents the GA search from being trapped a local optimum. A new individual is then created at the end of step 1. Step 2 (line 6 in Algorithm 4) also employs the repair mechanism (Algorithm 2). It corrects the new individual produced by the former step 1 with the “best effort”, if the environment cannot be covered at the required coverage rate at the end of step 1.

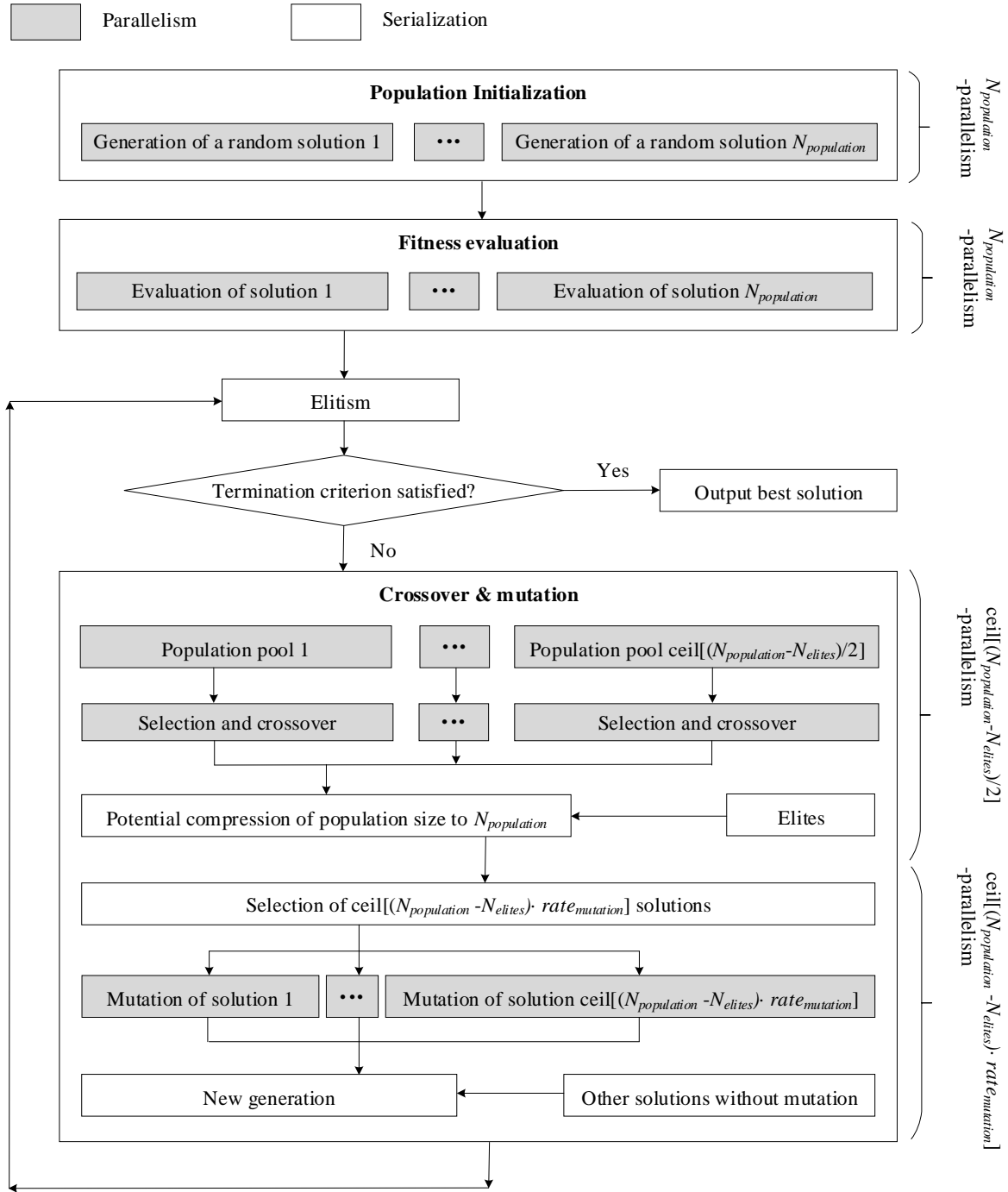


Fig. 5. Flowchart of the proposed parallel genetic algorithm

### 3.5 Parallel genetic algorithm

The efficiency of a GA search is sensitive to a large-scale optimization. A conventional GA structure can be found in [31]. The fundamental GA operations include: (1) population initialization, where a fixed size of individual solutions are generated in a random manner; (2) crossover, which swaps part of genes of two chromosomes (i.e., individual solutions); (3) mutation, which swaps genes (i.e., part of a solution) of a



chromosome; and (4) elitism, which remains a fixed size of the best individuals in a generation to the child generation.

All the aforementioned and fitness calculation of all individuals in a generation exhibit a common characteristic for applying “map-and-reduce” [33] or “divide-and-conquer” [15, 34, 35] parallel computation strategy: each GA operation contains multiple independent sub-operations of the same type and with different individuals. Therefore, the sub-operations can be conducted in parallel, such as by multithreads of a processor [36]. The results of sub-operations are then collected one by one at the end of each sub-operation. As a result, the GA search gains speedup as multithreads physically work in parallel in different cores of a processor.

Fig. 5 demonstrates the procedure of the proposed parallel GA for solving a large-scale TPC problem. Generally, parallelism is performed on all genetic operations that use Algorithm 2 to check and repair unqualified solutions: initialization (Algorithm 1), crossover (Algorithm 3), and mutation (Algorithm 4). The number of parallel operations depends on the peculiarity of a genetic operation type. In initialization, the random generation of  $N_{population}$  solutions are all independent.  $N_{population}$  RTPC (Algorithm 1) instances are thereby parallelized. To release the parallelism potential of crossover and mutation, both types of operations are performed based on a population instead of conventionally two solutions and one solution, respectively. For a generation, the numbers of parallel crossover and mutation are  $\text{ceil}[(N_{population}-N_{elites})/2]$  and  $\text{ceil}[(N_{population}-N_{elites}) \cdot \text{rate}_{mutation}]$ , respectively. Moreover,  $N_{population}$  fitness evaluations are parallelized at the end of a generation.

A special attention should be paid to the parallelism of multiple crossover operations. Normally, two individuals should be selected from the entire population for one crossover operation, in order to ensure that the better individuals have higher probability to be involved in breeding the child generation. As all the parallelized crossover operations should simultaneously have full access to the entire generation,  $\text{ceil}[(N_{population}-N_{elites})/2]$  population pools are created at the start of parallel crossover (Fig. 5). Each population pool is a copy of an entire population.

### 3.6 Additional speedup measures

As aforementioned, the design of GATPC in the former subsections follows the idea of decreasing computation time and memory, to enable large-scale optimization of TPC. The following measures are further taken to speed up the GATPC by reducing the computation redundancy.

An AP's maximal coverage distance ( $d_{j_{\max}}, \forall j \in J$ ) is extensively calculated by Algorithms 1-3. To speedup,  $d_{j_{\max}}$  is calculated by Eq. (8) before the actual start of a GA search. In total,  $N_p$  different  $d_{j_{\max}}$  values are pre-calculated according to  $N_p$  different AP transmit power values of an AP, and stored as a

constant vector. All the  $d_{j\max}$ -related calculation during the GA search process will then simply look up to this vector, instead of repeating the path loss calculation millions of times.

*GP-AP links* of all APs are very frequently established or removed in the correction procedure of Algorithms 1-3, which requires an extensive iteration of all possible GP-AP pairs ( $|\Omega| \cdot |A|$  in the worst case). This certainly becomes a tedious and time-consuming operation for a large environment that has more than 10,000 GPs as well as at least dozens or hundreds of APs. The corresponding speedup measure consists of the following four sequential steps. (1) For the  $j$ -th AP, search in the aforementioned  $d_{j\max}$  vector for its maximal coverage distance corresponding to the current transmit power level. (2) Set up a  $d_{j\max} \times d_{j\max}$  rectangular region that is centered at the  $j$ -th AP. (3) Iterate the GPs within this rectangular area and set up *GP-AP links* of the  $j$ -th AP. (4) Iterate all APs and conduct steps (1-3) in each iteration. The obtained speedup is especially significant for a large environment, since the area to set up *GP-AP links* is substantially reduced from the entire environment to the  $d_{j\max} \times d_{j\max}$  small square.

Besides, Algorithms 1-3 frequently judge whether an obstacle shadows the signal between the  $j$ -th AP and an Rx on the  $i$ -th GP, and then calculate the accumulated obstacle loss, i.e., Eq. (6, 7). The speedup measure is inspired from the fact that, for a certain GA search, all the obstacles and APs are static in terms of quantity and location. Consequently, the signal blockage between the  $i$ -th GP and the  $j$ -th AP can be judged before the GA search, and the corresponding obstacle loss (including zero loss) can be pre-stored in a table. The GA search will then only need to enquire the pre-stored table of obstacle loss by inputting the indexes of GP and AP, instead of on-the-fly judgment.

Last but not least, Algorithms 1-3 extremely frequently judge whether a GP is covered by an AP at its current transmit power level, i.e., Eq. (3). Thereby, the path loss calculation considering the shadowing effects of dominant obstacles should extensively be performed. As a speedup measure, Eq. (3) is implemented in the following sequential steps for the  $i$ -th GP and the  $j$ -th AP. (1) Look up to the aforementioned  $d_{j\max}$  vector for the corresponding  $d_{j\max}$  of the  $j$ -th AP. (2) Set up a  $d_{j\max} \times d_{j\max}$  square that is centered at the  $j$ -th AP. (3) Calculate the path loss between the  $j$ -th AP and the  $i$ -th GP, without considering the shadowing effects. (4) Look up to the aforementioned obstacle loss table, and add the obstacle loss to the path loss that is obtained in step (3), and get the final path loss value. (5) Obtain  $P_{ij}$  with the final path loss value and judge whether it is above the preset sensitivity threshold.

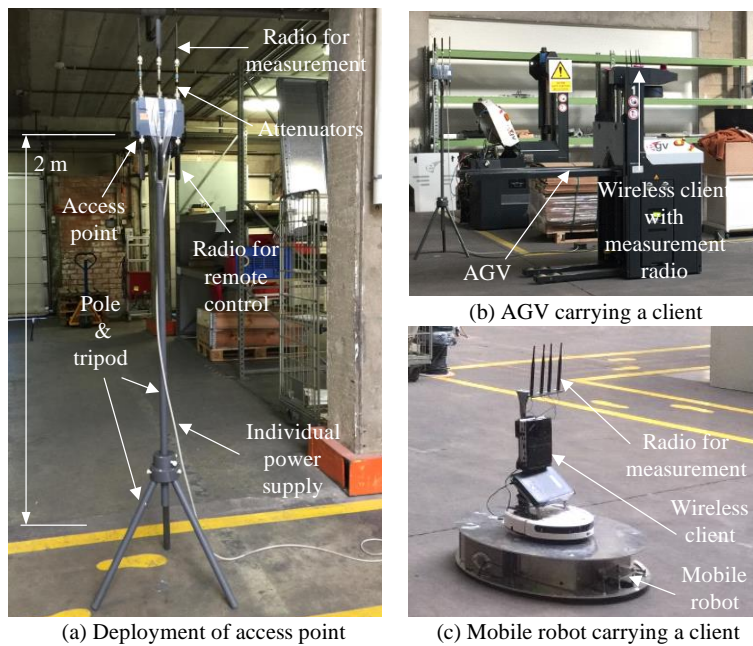


Fig. 6. Experimental facilities, including a measurement control computer system, four commercial off-the-shelf industrial access points, an automated guided vehicle (AGV) with a wireless client, and a mobile robot with a wireless client.

## 4. Experiment validation

The TPC model and the GATPC algorithm were validated in a small open industrial environment (10 m  $\times$  10 m) in the factory hall of a manufacturer of automated guided vehicles, in Flanders, Belgium.

### 4.1 Configurations

A measurement control system [1] accommodating the GATPC and four Siemens<sup>®</sup> industrial APs (Scalance W788-2 M12) with individual power supply (Fig. 6a) were used.

More specifically, an AP has two radio ports (Fig. 6a). One was configured for measurements at 2.4 GHz and the other was configured for remote control at 5 GHz, such that the interference between measurements and control is mitigated. For an AP, 44 dB attenuation was added to each of the three ports of the measurement radio, to mimic a larger environment needing four APs for double full coverage. Individual power supply plus an extension power cable was applied to every AP, to enable deployment without the distance limitation. The remote AP control was realized by SSH (secure shell). The central PC thus sent control wireless commands to an AP, such as setting the transmit power and powering on/off a radio.

The four APs were over-dimensioned on the boundary of the environment, such that each side was placed with one AP and double full coverage was planned [1]. The AP locations are indicated in Table 3, of which the coordinates are these used by the localization system of an automated guided vehicle.

The coverage measurement facilities that were used have been introduced in [1] in detail. They mainly include a measurement control software system, two Zotac<sup>®</sup> mini-PCs as two individual wireless clients, four

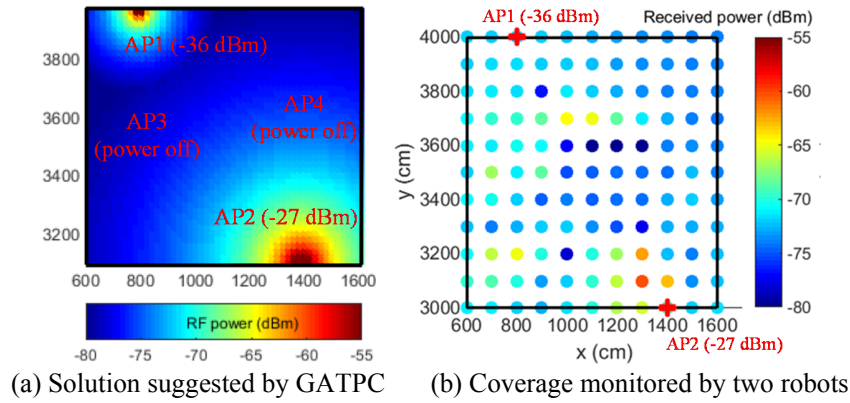


Fig. 7. Transmit power control solution given by the GATPC algorithm (Fig. 7a) and actual coverage monitored by an automated guided vehicle and a mobile robot which carry wireless clients (Fig. 7b). The simulated and measured coverage maps are highly matched.

**Table 3**

**Configurations of the measurement campaign**

AP transmit power range with attenuation	-39:1:-27 dBm
WLAN standard	IEEE 802.11n
AP working frequency band	2.4 GHz
AP remote control frequency band	5 GHz
AP height	2 m
AP1 location	(8, 40) m
AP2 location	(14, 31) m
AP3 location	(6, 35) m
AP4 location	(16, 36) m
Required physical bitrate of a wireless client	24 Mbps
Required receiving sensitivity	-79 dBm
Mobility speed of the automated guided vehicle and robot	20 cm/s
Grid size for coverage monitoring	1 m
Shadowing margin (95%)	1 dB
Fading margin (99%)	0 dB
Interference margin	0 dB
GATPC stop criterion	30 iterations

poles with tripods to support the APs at the height of 2 m (Fig. 6a), an automated guided vehicles as a controllable mobile vehicle which carries one client on the top (Fig. 6b), a w-iLab.t mobile robot [37] which carries the other client on the top (Fig. 6c).

Instead of manual measurements, the two clients automatically kept on moving around in the environment and measuring the coverage of the AP that they connected to, and fed the collected samples back to the central PC for monitoring. These samples were stored in database of the measurement control system. Samples from the same AP and within the same spatial grid were further aggregated to one value (dBm) to enable stable coverage monitoring (Sect. 2.1). For the minority of grids that might contain no sample, interpolation [1, 38] was applied based on the surrounding samples. Table 3 lists the key measurement configurations.

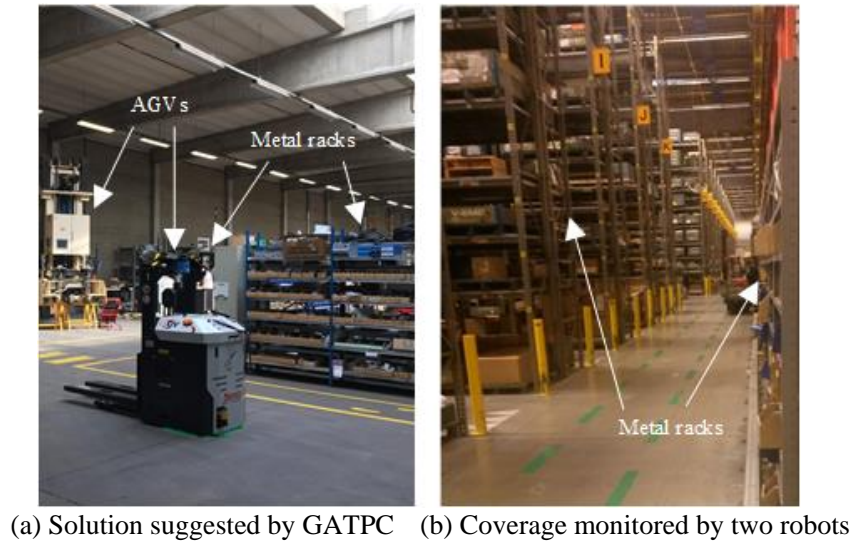


Fig. 8. The two industrial indoor environments for numerical experiments: a factory hall of an automated guided vehicle (AGV) and a warehouse of a car manufacturer. Both environments are placed with metal racks, which creates a challenge for radio propagation or robust wireless connection.

In total, 3745 RF power samples were collected. Regression [3] was applied to these data to build an empirical path loss model formulated by Eq. (5), where  $PL0$  was 39.87,  $n$  was 1.78, and the obstacle loss  $OL_{ij}$  ( $\forall i \in I, \forall j \in J$ ) was zero dB due to the empty environment. The R-squared value was 97.38%, indicating that the path loss model was highly fitted to the samples.

#### 4.2 Validation results

The TPC solution given by the GATPC algorithm is illustrated by Fig. 7a. AP1 and AP2 are powered on at -36 dBm and -27 dBm, respectively. AP3 and AP4 are both powered off. The colored GPs (grid points) represent the highest received RF power from the existing APs. All the received RF power values are above the required lowest sensitivity (-79 dBm, Table 3), indicating one full coverage layer in the environment. In a conventional full power-on scheme, all the four APs are simply powered on with the maximal transmit power (-27 dBm). In comparison, in the obtained TPC solution, AP1 decreases the transmit power to -36 dBm, and AP3 and AP4 are powered off.

The power states and transmit power levels of the four deployed APs were then set according to this optimal TPC solution. The coverage was monitored. As visualized in Fig. 7b, the received RF power values vary between -60.4 dBm and -78.8 dBm. They are above the threshold sensitivity (-79 dBm, Table 3), demonstrating that the environment is fully covered. Therefore, the solution given by GATPC are effective to satisfy the major constraint (i.e., coverage, Eq. (13)) of the TPC model.

## 5. Numerical experiments

Numerical experiments were further conducted on the proposed GATPC algorithm. A 64-bit Win7 PC was used, with an Intel i5-3470 CPU and an 8 GB RAM.

### 5.1 Configurations

The two investigated industrial indoor environments are a factory hall of an automated guided vehicle manufacturer and a warehouse of a car manufacturer, both located in Flanders, Belgium.

The automated guided vehicle factory hall (Fig. 8a) measures  $102\text{ m} \times 24\text{ m}$ . It represents a small-scale industrial indoor environment. It is placed with metal racks for component storage. Vehicles of varying sizes are usually placed and waiting for integration, maintenance, or shipment. Wide WiFi coverage is needed for

**Table 4**  
**Configurations of the numerical experiments**

		Path Loss Model
PL0		39.87 dB
$n$		1.78
Shadowing margin (95%)		7 dB
Fading margin (99%)		5 dB
Interference margin		0 dB
		Access Point (AP)
Height		2 m
Gain		3 dB
WiFi standard		IEEE 802.11n
Transmit power range		$\{-5:1:7\}$ dBm
Locations		Output by over-dimensioning
Numbre of APs		4 (small-scale environment) 75 (large-scale environment)
		Wireless Client
Height		1.4 m
Gain		2.15 dB
Required physical bitrate		54 Mbps
Required minimal sensitivity		-68 dBm
		Environment
Factory hall	Size (small scale)	$2448\text{ m}^2$ ( $102\text{ m} \times 24\text{ m}$ )
	Grid point number	2600
Warehouse	Size (large scale)	$83,000\text{ m}^2$ ( $415\text{ m} \times 200\text{ m}$ )
	Grid point number	83,616
Grid size ( $gs$ )		1 m
Radio frequency		2.4 GHz
Antenna type		Omnidirectional
Metal rack size		$20\text{ m} \times 3\text{ m} \times 9\text{ m}$
Path loss caused by one metal rack		7.37 dB
		GATPC algorithm
Population size		60 (small-scale environment), 100 (large-scale environment)
Elitism rate		4%
Crossover rate		70%
Mutation rate		40%
Stop criterion		50 iterations

vehicle communication and Internet access of the workers' laptops.

The warehouse (Fig. 8b) measures  $415 \text{ m} \times 200 \text{ m}$ . It represents a large-scale industrial indoor environment. It is placed with metal racks at a height of nine meters. These racks are filled with wooden boxes that contains metal components. Wide WiFi coverage is required to support voice picking. Human pickers are equipped with microphones and earphones. They communicate with the control center via WLANs, to pick up from and place a stuff to a specific location.

For the TPC model, a metal rack in both cases is an obstacle that potentially causes evident shadowing effects to radio propagation. In the following numerical experiments, an obstacle measures  $20 \text{ m} \times 3 \text{ m} \times 9 \text{ m}$ . It can be placed either horizontally (i.e., the length side is parallel to the length side of the environment) or vertically (i.e., the length side is parallel to the width side of the environment). The direction and location of an obstacle are randomly generated by following a uniform distribution, while the entire part of a rack must be enclosed in the environment. The number of racks is an input of the TPC model. The GPs occupied by obstacles are not considered in the path loss calculation.

The network parameters are summarized in Table 4, including the path loss model, AP Tx, Rx, and environment. APs deployed by using the over-dimensioned algorithm such that two full coverage layers are created in the target environment [1]. Each AP has 14 different transmit power levels, including powering off.

As pointed out in [15], the grid size ( $gs$ ) influences the computational accuracy of coverage,  $gs$  should be as small as possible without significantly compromising the computational complexity. Consequently,  $gs$  is set as one meter, which is within 10 wave length (1.2 m). This means that the path loss within this distance can be considered as constant without sacrificing the precision of path loss calculation. The two parameters  $PL0$  and  $n$  of the one-slope path loss model are same as these in Sect. 4. The path loss caused by a metal rack (7.37 dB) is the mean of measured path loss samples. The GA parameters are shown in Table 4. The stop criterion is 50 iterations, during which the GA was found to usually stagnate.

Compared with the two environment sizes ( $68 \text{ m} \times 59 \text{ m}$  and  $12 \text{ m} \times 67 \text{ m}$ ) and around 30 APs involved in large-scale WLAN design in [17], our investigated two environments, especially the warehouse environment, show their hyper-large property for optimization.

### 5.2 Effectiveness in empty environments

The GATPC was first performed in the small-scale and large-scale environments without any presence of metal obstacles while one full coverage layer was guaranteed ( $\mu = 1$ ). Two other transmit power management schemes were used for benchmarking. One is the RTPC scheme (Algorithm 1). The other is the full power-on scheme, where all APs are powered on with maximal transmit power, i.e., no TPC is deployed.

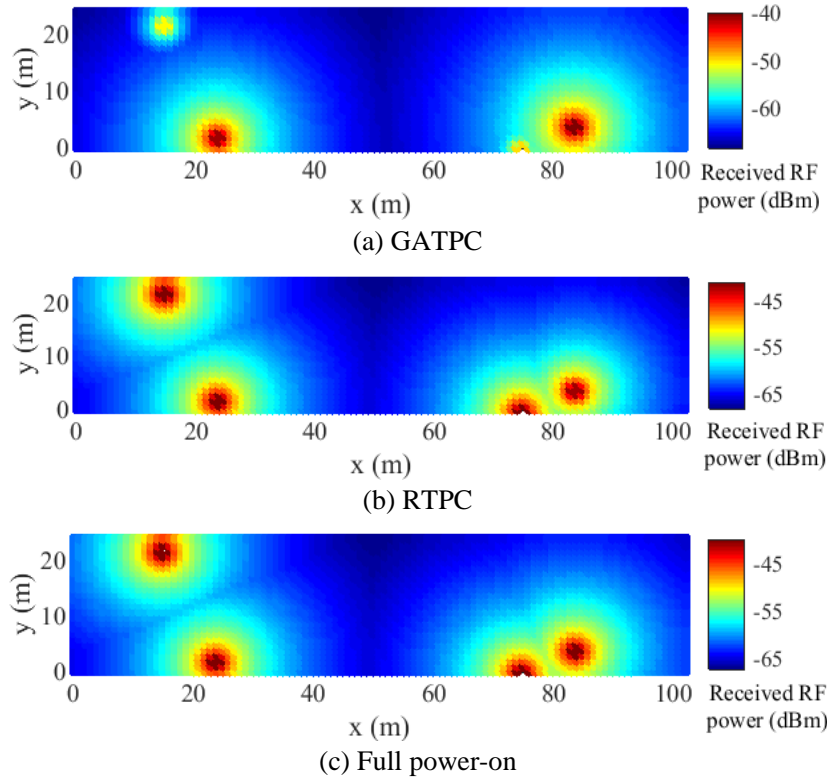


Fig. 9. Three transmit power control schemes for an empty small-scale environment. The proposed GATPC algorithm is notably superior in reducing the transmit power or coverage of over-dimensioned wireless nodes while ensuring full coverage in the environment.

### 5.2.1 Small-scale empty environment with full coverage

Overall, the GATPC is demonstrated to have notable superiority over the two benchmark schemes, in terms of reducing transmit power of wireless nodes and minimizing total interference in the network. More detailed results will be described as follows.

The GATPC significantly decreases the coverage of two APs while ensuring one full coverage layer in the environment. The transmit power of the four over-dimensioned APs is -4 dBm, 6 dBm, -3 dBm and 7 dBm, respectively, from the left to the right of Fig. 9a. In contrast, the RTPC exhibits very limited performance in reducing the redundant transmit power. Its coverage map (Fig. 9b) is close to that of the full power-on scheme (Fig. 9c). Its suggested transmit power is 6 dBm, 6 dBm, 6 dBm and 5 dBm, respectively.

As indicated in Table 5, the GATPC evidently reduces the total interference (-32.04 dBm). In comparison, the RTPC shows limited capacity in mitigating interference. Its interference level (-24.95 dBm) is close to that produced by the worst case (-23.83 dBm in full power-on scheme).

Besides, the runtime of GATPC is short (64 sec, Table 6). The RTPC has nearly zero runtime (Table 5), since optimization is not involved and the environment is small.



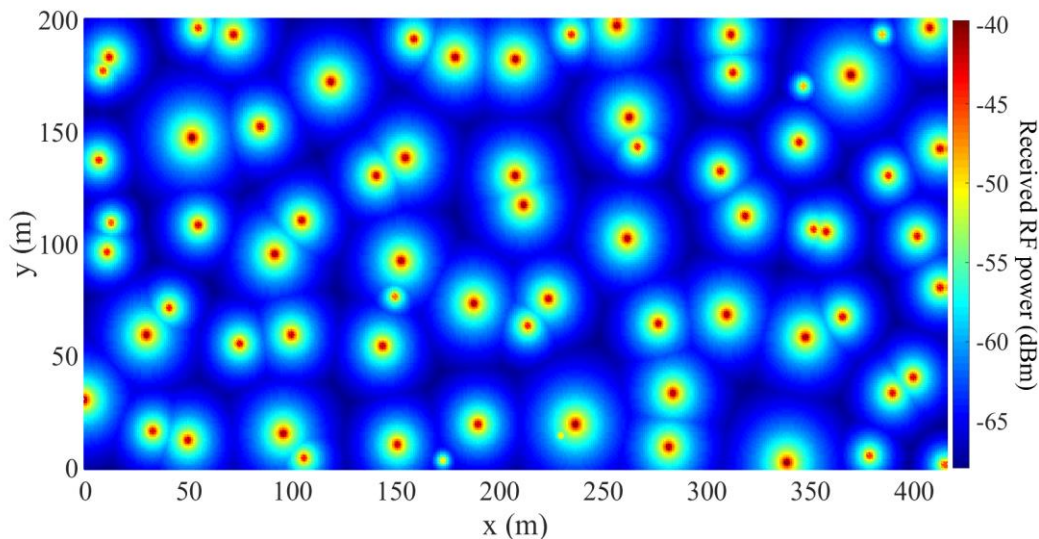


Fig. 10. Transmit power control solution suggested by the GATPC algorithm for an empty large-scale environment. Among the 75 over-dimensioned APs, 4 are powered off and most are set by transmit power lower than the maximum, while still having full coverage.

### 5.2.2 Large-Scale Empty Environment with Full Coverage

The GATPC exhibits superior interference minimization performance in the large-scale empty environment. It achieves an interference level of -9.71 dBm in comparison to -9.29 dBm in the RTPC scheme and -7.02 dBm in the full power-on scheme (Table 5).

The GATPC is effective in AP transmit power reduction (Fig. 10). Besides the four powered-off APs, most of the powered-on APs are set by a transmit power level that is lower than the maximum (7 dBm, Table 4), and many are even set by a level which is very close or equal to the minimum (-5 dBm, Table 4).

The GATPC's runtime obviously increases (102,841 sec or about 28.5 h, Table 6) compared with that in a

**Table 5 Interference of different transmit power control (TPC) schemes and runtime of RTPC**

Environment type		Small empty	Small obstructed	Large empty	Large obstructed
GATPC	Interference (dBm)	-32.04	-26.59	-9.71	-10.02
	Runtime (s)	0	0	103	131
RTPC	Interference (dBm)	-24.95	-24.97	-9.29	-9.56
	Runtime (s)	0	0	103	131
Full power-on	Interference (dBm)	-23.83	-24.04	-7.02	-7.52

**Table 6 Speedup performance of the GATPC algorithm using high performance computing (HPC)**

Runtime	Small empty	Small obstructed	Large empty	Large obstructed
With HPC (s)	64	73	102,841	167,504
Without HPC (s)	94	172	3,866,700	4,670,300
Reduction rate (%)	31.9	57.6	97.3	96.4
Speedup times	0.5	1.4	37.6	27.9

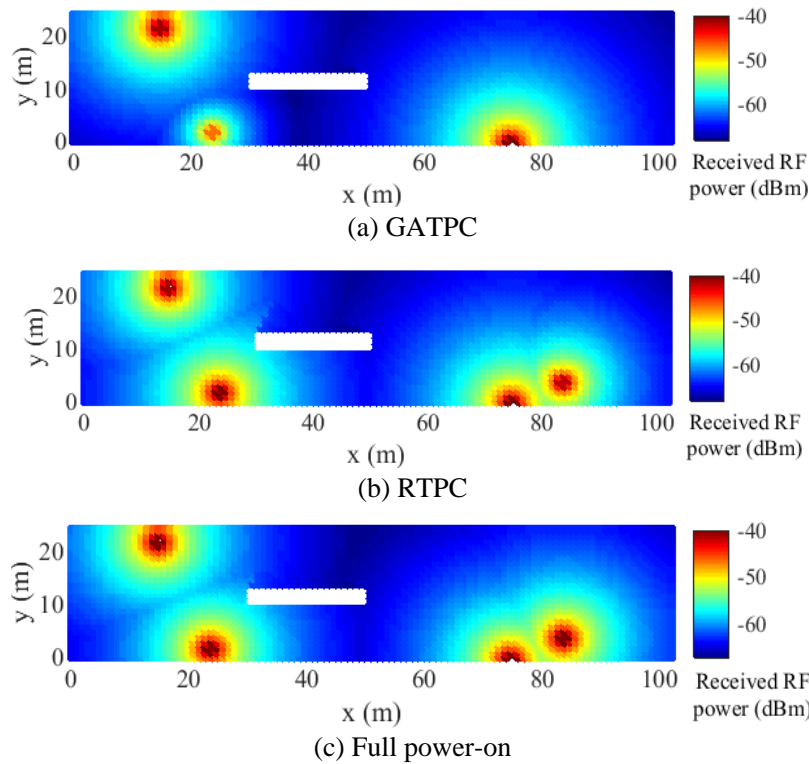


Fig. 11. Three transmit power control schemes for a small-scale environment placed with a metal rack. The proposed GATPC algorithm is evidently superior in reducing the transmit power or coverage of over-dimensioned wireless nodes while ensuring full coverage in the environment.

small-scale empty environment (64 s, Table 6). This is explained by the 34 times larger area and the consequently 603 times more AP-GP pairs in the large-scale environment.

### 5.3 Effectiveness in Obstructed Environments

The GATPC was then performed in these small-scale and large-scale environments which are obstructed while one full coverage layer is still guaranteed ( $\mu = 1$ ). To mimic the shadowing effects in industrial indoor environments, one metal rack (Table 4) was placed in the small-scale environment and ten in the large-scale environment with a 100% qualification rate. The two aforementioned benchmark schemes were also used to measure GATPC's performance.

#### 5.3.1 Small-Scale Obstructed Environment

The GATPC obviously demonstrates superior TPC effectiveness in the small-scale obstructed environment. According to its output solution, it not only powers off one of the four APs, but also decreases the other two's transmit power (0 dBm and 6 dBm) while keeping the fourth one at the maximum (Fig. 11a).

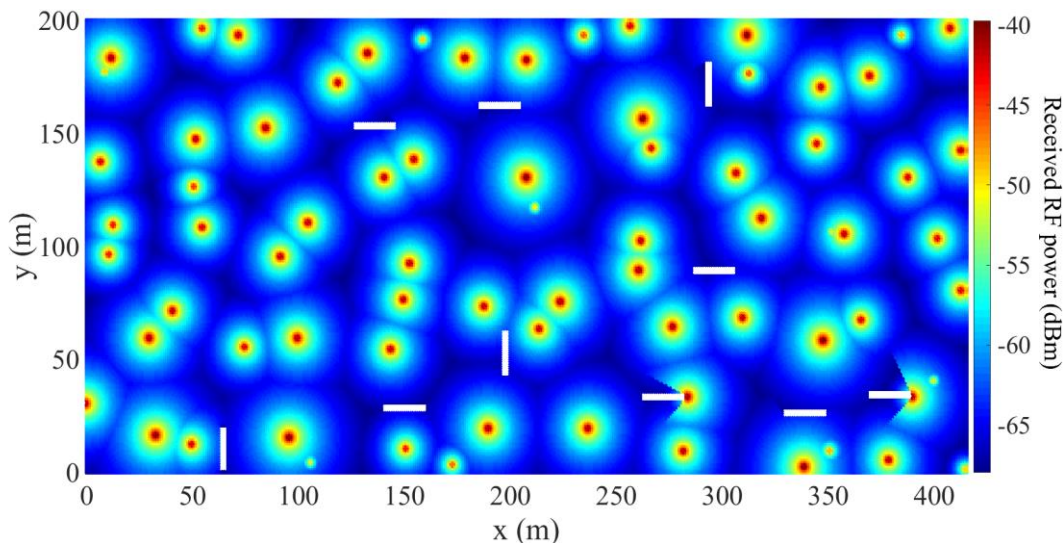


Fig. 12. Transmit power control solution suggested by the GATPC algorithm for an obstructed large-scale environment (the 10 white rectangles represent 10 randomly placed metal racks). Besides one AP that is powered off, many of the rest APs reduce their transmit power close to the minimum, while still ensuring full coverage.

In contrast, the RTPC scheme exhibits little capacity to reduce the transmit power. Its output TPC solution (Fig. 11b) is quite close to that of the full power-on scheme (Fig. 11c).

The GATPC also shows up as the best in interference mitigation in the small-scale obstructed environment. It suppresses the inference down to -26.59 dBm (Table 5). This is lower than -24.97 dBm in the RTPC scheme and -24.04 dBm in the full power-on scheme (Table 5).

GATPC's runtime is short (73 s, Table 6), due to the small scale of the investigated environment. It slightly increases compared to that in the small-scale empty environment. This is because of the additional obstacle loss calculation (Eqs. (6-7)), though the GPs taken up by the metal rack are excluded in the interference calculation. For the same two reasons (Sect. 5.2.1), the RTPC scheme has almost zero runtime.

### 5.3.2 Large-Scale Obstructed Environment

The GATPC also exhibits superiority in interference mitigation in the large-scale obstructed environment. It achieves a total interference level of -10.02 dBm, while the RTPC and full power-on schemes produce interference of -9.56 dBm and -7.52 dBm, respectively (Table 5).

The effectiveness of GATPC in TPC is further demonstrated in Fig. 12, which presents the corresponding coverage map. One AP is powered off and three APs are powered on with the minimal transmit power of -5 dBm. Among the APs that are powered on, many have transmit power levels that are lowered close to the minimum.

For the same two reasons explained in Sect. 5.2.2, the runtime of the GATPC rises to 167,504 s compared to 73 s in the small-scale obstructed environment (Table 6). Due to the additional obstacle loss calculation, it is also larger than that in the large-scale empty environment (102,841 s, Table 6).

### 5.4 Effectiveness in Speedup

To further benchmark the GATPC's speedup performance, a derived version was used. It is the GATPC without high performance computing (HPC), including the parallel processing (Sect. 3.5) and speedup measures (Sect. 3.6). As it turned out to be very time-consuming to obtain an optimal solution in the large-scale environment (at the unit of months), the following means was taken to gauge its runtime.

First, the runtime to generate one random solution in the initial population was measured (at the scale of thousands of seconds). It was then multiplied by the population size to get the total runtime for population initialization. This derived GATPC version was rerun by enabling HPC in population initialization (Sect. 3.2) and followed by population evolution (Sects. 3.3 & 3.4) without HPC. Once the population went through one evolution and its corresponding runtime was got (at the scale of ten thousands of seconds), this algorithm stopped and this runtime was multiplied by the number of evolutions to obtain the runtime for evaluating the entire population. Finally, the estimated overall runtime was the sum of the runtime for the population initialization and that for the population evolutions.

The GATPC with HPC demonstrates significant speedup performance, as presented in Table 6. In the small-scale environment, its speedup times stay around one. The runtime of both algorithms are acceptable. However, in the large-scale environment, the speedup times boost to around 30. This makes it feasible to run the GATPC algorithm in a dramatically-reduced time horizon (1 - 2 days), in contrast to the infeasible runtime of the version with HPC. Given that a factory's major layout cannot change too frequently, this optimized runtime is acceptable from the perspective of adapting TPC to a factory layout while minimizing the network interference.

### 5.5 Sensitivity of Qualification Rate

As the "best effort" philosophy (Sects. 3.2-3.4) is applied in the GATPC algorithm, the qualification rate

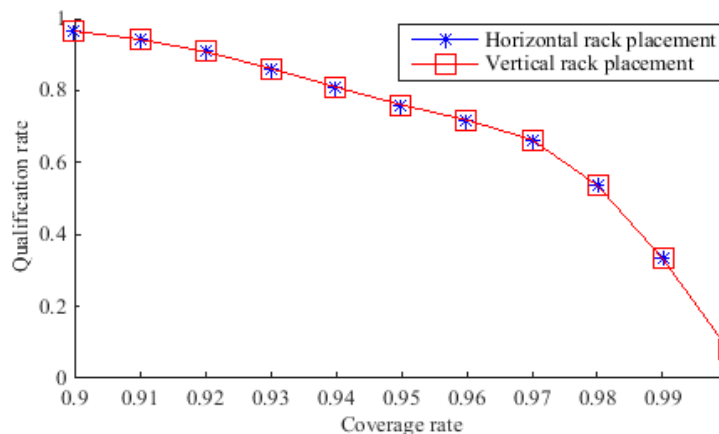


Fig. 13. Transmit power control (TPC) qualification rate to satisfy the required coverage rate in a metal-dominating environment. For each coverage rate, the rack location iterates over all the possible grid points with horizontal and vertical placement direction. As shown, 90% coverage can be guaranteed by the GATPC algorithm in more than 95% shadowing cases.

is investigated. This rate means the probability for this algorithm to intrinsically satisfy the TPC model's fundamental constraint (i.e., coverage, Eq. (13)) when all APs are powered on with the maximal transmit power. During this experiment, the correspondent interference was calculated each time when one metal rack was shifted to a different GP in the small-scale environment. This calculation was iterated over all the possible GPs. Consequently, the relationship between this qualification rate and the required coverage rate was captured.

As depicted in Fig. 13, 90% coverage can be guaranteed in more than 95% shadowing cases, demonstrating the GATPC's effectiveness in a general obstructed environment. The qualification rate is insensitive to the placement direction of a metal rack. It achieves as high as 96.5% at the coverage rate of 90%. It gradually decreases with the rising coverage rate, and finally drops to 8.1% in the case of full coverage. This decrease is explained by some specific rack locations on which a rack shadows all the potential over-dimensioned APs for some specific GPs. If a coverage level higher than 90% is desired for 95% shadowing cases, this improvement would rely on the over-dimensioning algorithm, instead of the TPC algorithm.

### 5.6 Sensitivity of Interference

The correlation between the interference and required coverage rate was further investigated under a varying number of metal racks placed in the small-scale environment. For each configuration, 30 independent runs were conducted and the average interference was collected, in order to get representative optimization results.

As indicated in Fig. 14, the interference declines from about -30 dBm at full coverage to -37 dBm at 50% coverage, regardless of the number of metal racks. This insensitivity to the number of metal racks implies that the limited number of GPs occupied by metal racks does not contribute much to the overall interference.

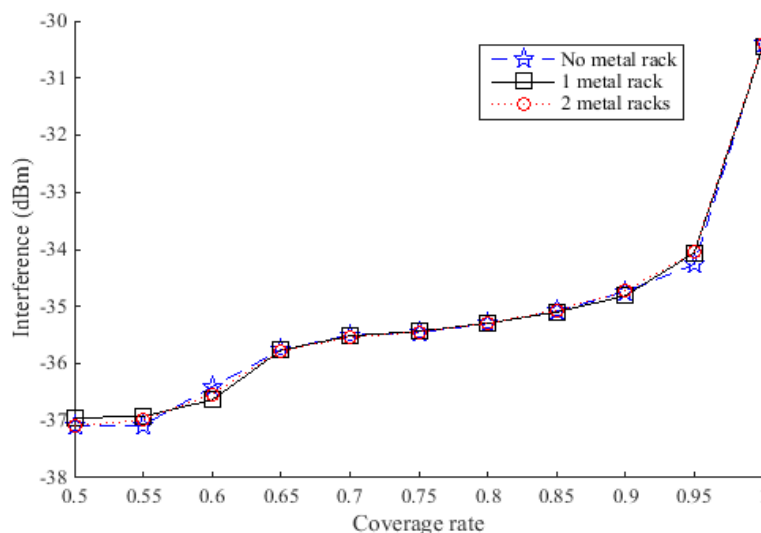


Fig. 14. Overall network interference under varying coverage rate and number of metal racks

This drop is explained by the continuously decreased AP transmit power to satisfy the TPC model's coverage constraint which consistently becomes less strict. This is further demonstrated by the optimization results, where the number of APs that are powered off generally increases with the reduction of required coverage rate.

Furthermore, the 10% coverage reduction from 100% to 90% contributes to more than 60% of the overall decreased interference (Fig. 14). This implies that lowering the required coverage rate cannot be highly effective when the desired coverage rate drops below 90%. When the coverage falls in the range between 90% and 65%, the interference nearly remains stable. This is further proved by the optimization results, where the number of APs that are powered off almost remains 1 when the coverage declines from 95% to 65%. In spite of the slight decrease in interference when the coverage continues to drop from 65% to 50% (Fig. 14), the seriously-affected coverage should dramatically overweight this gentle decrease. Therefore, when using GATPC, a coverage rate between 90% and 100% not only guarantees a high coverage level for wireless clients, but also is an effective range to control the overall interference.

### 5.6 Performance Comparison with Benchmark Algorithms

Besides the demonstrated effectiveness and scalability of the proposed GA, its performance is further compared with other state-of-the-art algorithms. The benchmark algorithms include (1) the proposed GATPC, (2) the proposed GA with the Boolean disk model which is commonly used in literature (GABD) [14, 15], (3) the proposed GA without repair mechanism (GAW), (4) a discrete version of particle swarm optimization (PSO) algorithm [39], and (5) an ant colony optimization (ACO) algorithm [40] which were used in two similar wireless coverage problems, as well as (6) the former PSO with the repair mechanism proposed in this paper (Algorithm 2), which is named PSOR.

Only the option of powering on/off APs is enabled in the GABD, due to the Boolean disk model. As no repair mechanism is used in GAW, PSO, and ACO, a solution is assigned the maximal fitness if it does not satisfy the required coverage rate, aiming to eliminate unqualified solutions. To get tailored for the TPC problem, the ACO in [40] has the following adaptations. The construction graph in Fig. 3 of [40] is a  $(N_p + 1) \times |A|$  matrix. An ant goes from the leftmost to the rightmost column and selects one vertex in each column. This constructs a TPC solution. The upper bound of the solution ( $C$ ) is the interference when all APs are powered on. The construction rule guides an ant to select the transmit power level of an AP. The pheromone is deposited between two APs and is calculated by Eq. (10) in [40]. The heuristic information is based on the increment in the actual coverage rate. The probability to select a transmit power level is calculated by Eq. (12) in [40], except that the index  $k$  refers to a transmit power level of an AP. An ant selects a transmit power level using Eq. (13) in [40]. The pheromone is updated by Eq. (15) in [40]. The local search



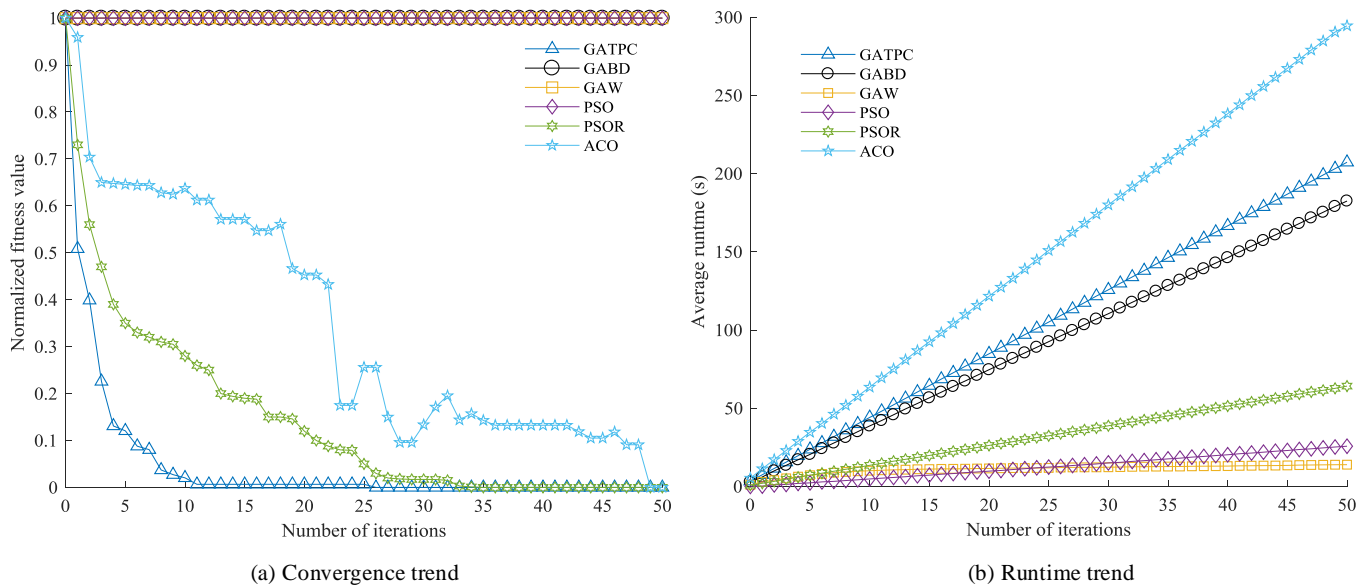


Fig. 15. Convergence and runtime trends of six optimization algorithms in a small obstructed environment (both trends are averaged over 30 independent runs)

procedure is similarly to check and power off redundant APs after a best-so-far solution is updated at the end of each iteration.

For all algorithm instances, the required coverage rate is 1; the maximal iteration is 50; the other configurations remain these in the original literature. 30 runs and 1 run of each algorithm instance are performed for a small and large problem size, respectively, in order to evaluate the optimization efficiency and scalability effectiveness of these algorithms, respectively. The performance comparison among these 6 algorithms is conducted in three dimensions: the interference produced the optimal TPC solution, the runtime of an algorithm, and the percentage of runs that comply with the required coverage rate. An algorithm instance terminates if it has a runtime longer than 48 h.

Fig. 15 further presents the convergence and runtime trends of these six algorithms in a small obstructed environment, both of which are averaged over 30 runs to have statistical significance. As demonstrated by Fig. 15a, GATPC achieves the fastest convergence by rapidly dropping during the first 5 generations, nearly staying stable after the 10<sup>th</sup> generation, and slightly decreasing at the 26<sup>th</sup> generation. PSOR has a slower convergence trend compared to GATPC by steadily dropping until the 34<sup>th</sup> iteration. ACO has a relatively unstable convergence trend. It fast drops in the first 4 iterations, and stays on a plateau with a gentle decrease until the 22<sup>nd</sup> iteration. This process iterates until the 49<sup>th</sup> iteration. A reason for this phenomenon could be that ACO cannot fully get stable in 50 iterations, revealing its relatively weaker search competence compared to GATPC and PSOR.

**Table 7 Performance comparison of six optimization algorithms for the transmit power calibration (TPC) problem in small and large-scale obstructed environments (best performance is in bold)**

Algorithm	Small obstructed environment <sup>a</sup>			Large obstructed environment <sup>b</sup>		
	Interference (dBm)	Runtime (s)	Coverage requirement compliance	Interference (dBm)	Runtime (s)	Coverage requirement compliance
GATPC	-26.51 ± 0.11	207 ± 4	<b>100%</b>	-10.02	167,504	<b>100%</b>
GABD	-25.28 ± 0	182 ± 2	<b>100%</b>	n.a. <sup>c</sup>	> 187 h	n.a. <sup>c</sup>
GAW	-27.80 ± 2.54	<b>14 ± 5</b>	85%	-9.72	172,563	100%
PSO	<b>-27.83 ± 3.86</b>	20 ± 3	87%	<b>-12.85</b>	<b>45,170</b>	72%
PSOR	-26.52 ± 0.12	80 ± 5	<b>100%</b>	-10.24	105,070	<b>100%</b>
ACO	-25.47 ± 0.07	290 ± 8	<b>100%</b>	n.a. <sup>c</sup>	> 600 h	n.a. <sup>c</sup>

<sup>a</sup>: mean ± standard deviation are used to evaluate the performance of 30 independent runs

<sup>b</sup>: the optimal solution obtained by one run is used due to the long runtime required by a large-scale TPC problem

<sup>c</sup>: unavailable performance data due to a runtime longer than 48 h

On the other hand, GABD, GAW, and PSO cannot effectively improve the best solution across iterations by having constant fitness value of one in Fig. 15a. This phenomenon is explained by two reasons. For GABD, this would be caused by a lack of fine-tuned transmit power control mechanism, such that the best solution in the initial population cannot be further enhanced by only powering on/off APs. For GAW and PSO, this is due to the missing repair mechanism for unqualified solutions. Consequently, a population cannot make effective progress toward better solutions through iterations.

In terms of average runtime trends of these six algorithms (Fig. 15b), all algorithms have a runtime which linearly increases with the number of iterations, except GAW of which the rise of runtime rises slows after the 16<sup>th</sup> iteration. This common linear increase in runtime is explained by the fact that the number and type of operations performed in each iteration of these optimization algorithms are more or less the same. ACO has the most rapidly-increasing runtime, which is evidently longer than that of other algorithms in each iteration. PSO and GAW have the most slowly-rising runtime before and after the 25<sup>th</sup> generation, respectively. GATPC has a runtime which increases slightly faster than that of GABD. This implies that the consideration of fine-tuned transmit power control does not obviously increase the computational burden of an optimization algorithm. However, the consideration of repair mechanism for unqualified solutions evidently slows down the evolutionary search, by comparing the runtime of GATPC and GAW in Fig. 15b. It also moderately slows down the particle swarm search, by comparing the runtime of PSOR and PSO (Fig. 15b). This highlights the computational overhead of repairing unqualified solutions during an optimization process.

Table 7 tabulates the performance comparison of these six algorithms on both small and large problem sizes, regarding interference minimization, runtime, and compliance with full coverage requirement. Although GAW and PSO produce solutions with the lowest mean interference (-27.80 dBm and -27.83 dBm) in a small obstructed environment, they cannot fully satisfy the required coverage rate and lead to the highest



variance in interference (2.54 dBm and 3.86 dBm, which are both at least one order of magnitude larger than that variance in interference of other algorithms). On the other hand, GATPC and PSOR, both using the proposed repair mechanism (Algorithm 2), achieve the lowest mean interference (-26.51 dBm and -26.52 dBm) among algorithms that 100% comply with the coverage requirement as well as low variance in interference (0.11 dBm and 0.12 dBm). This comparison underlines the undeniable importance of the proposed repair mechanism in reducing ineffectiveness of a search. Comparatively, assigning maximal fitness values to unqualified solutions is not an effective method to search for qualified solutions in the solution space.

Similarly, due to a lack of repair mechanism, GAW and PSO have the shortest mean runtime (14 s and 20 s) of on a small scale (Table 7), respectively, while breaking the hard constraint of required coverage percentage. GATPC achieves both moderate average (207 s) and variance (4 s) in runtime on a small scale. A remarkable observation is on PSOR. It has a mean runtime evidently shorter than that of GATPC (80 s vs. 207 s), while remaining a superior level in interference and full coverage requirement compliance. Conversely, the mean interference and runtime of ACO are inferior on a small scale (-25.47 dBm and 290 s), though it can effectively eliminate unqualified solutions by assigning maximal fitness values to unqualified solutions. A reason would be that an ACO-based algorithm highly needs a tailored design for a specific problem (e.g., transmit power control in this paper), in terms of construction graph, construction rule, pheromone management, and local search procedure. These observations of runtime in a small-sized problem in Table 7 comply with the runtime trend depicted in Fig. 15b.

The contribution of the proposed path loss model or coverage calculation method is also underlined by comparing GABD to GATPC in the small obstructed environment (Table 7). As only the option of powering on/off is available in GABD, the produced mean interference is the highest (-25.28 dBm). Its zero variance in interference is also explained by a lack of fine-tuned transmit power control mechanism such that the local optimum can be easily found by randomizing a number of solutions in the initial population and no better solution is found through the evolutionary search. This reason is further justified by Fig. 15a, where GABD cannot improve the best solution in the initial population with the rising number of generations.

The former observations remain in the large obstructed environment (Table 7) except the following two points. Firstly, though the mean runtime of GABD (182 s) is comparable to that of GATPC (207 s) for a small problem size, this gap evidently rises ( $> 187$  h vs. 167,504 s, i.e., more than 4-times difference) for a large problem size. This should be due to the characteristic of the proposed repair mechanism. Once an AP is powered on, it achieves the maximal transmit power, losing the potential to collaborate with other APs for further coverage. This thus triggers lines 18-20 in Algorithm 2 much more frequently to remove the invalid *GP-AP links* due to a loss of capability to fine-tune the coverage. Secondly, ACO is still the slowest

optimization algorithm ( $> 600$  h). But it has an even larger runtime gap with other algorithms. This indicates that ACO especially requires tailored design for large-scale optimization despite the common challenge faced by the other algorithms to work on a large problem size.

## 6. Conclusion

With the ongoing trend toward factories of the future, wireless technologies are penetrating to shop floors and warehouses, which include not only wireless sensors networks (WSNs) but also wireless local area networks (WLANs). This paper formulates a transmit power control (TPC) model for dense industrial WLAN (IWLANS). It addresses the drawbacks of existing coverage-related optimization models, by focusing on scalability, simple yet accurate coverage prediction considering three-dimensional shadowing effects in harsh industrial indoor environments, complete power management schemes (with both powering on/off and transmit power calibration mechanisms), and empirical validation. To solve this TPC model, this paper proposes genetic algorithm based TPC (GATPC). Repair mechanism-based population initialization, crossover and mutation are designed to reduce the GA search redundancy. Parallelism and dedicated speedup measures are further proposed to speed up a GATPC instance for both small- and large-scale optimization.

The GATPC was experimentally validated with a real IWLANS deployed in a small-scale industrial environment, and numerically demonstrated in both small and large problem sizes. The solution quality of the GATPC was proven in terms of effectively conducting adaptive coverage and minimizing interference even in the presence of metal obstacles. The speedup performance of GATPC was measured to be as high as 37 times compared to the serial GATPC without speedup measures. The effectiveness and scalability of GATPC was further demonstrated by comparing to other state-of-the-art algorithms. In sensitivity studies, the produced interference and qualification rate of GATPC are revealed according to varying required coverage rate as well as number and placement direction of dominant obstacles.

The formulated TPC problem and the proposed GATPC algorithm can also be applied to other types of wireless network besides WLANs, e.g., optimal coverage maintenance of WSNs [20] as well as RFID network planning and configuration [41]. Regarding the future work, further speedup measures or high-performance algorithm design paradigms may be explored to additionally reduce the runtime of the GATPC.

## Acknowledgement

This research was supported by the ICON-FORWARD project with imec.

## References

- [1] X. Gong, J. Trogh, Q. Braet, E. Tanghe, P. Singh, D. Plets, J. Hoebeke, D. Deschrijver, T. Dhaene, L. Martens, and W. Joseph, "Measurement-based wireless network planning, monitoring, and reconfiguration solution for robust radio communications in indoor factories," *IET Science, Measurement & Technology*, vol. 10, no. 4, pp. 375-382, 2016.
- [2] G. Cena, L. Seno, A. Valenzano, and C. Zunino, "On the performance of IEEE 802.11e wireless infrastructures for soft-real-time industrial applications," *Industrial Informatics, IEEE Transactions on*, vol. 6, no. 3, pp. 425-437, 2010.
- [3] E. Tanghe, W. Joseph, L. Verloock, L. Martens, H. Capoen, K. V. Herwegen, and W. Vantomme, "The industrial indoor channel: large-scale and temporal fading at 900, 2400, and 5200 MHz," *IEEE Transactions on Wireless Communications*, vol. 7, no. 7, pp. 2740-2751, 2008.
- [4] J. Tang, C. Zeng, and Z. Pan, "Auction-based cooperation mechanism to parts scheduling for flexible job shop with inter-cells," *Applied Soft Computing*, vol. 49, pp. 590-602, 2016.
- [5] S. Roswitha, "A wireless future for industry," Siemens, 2014; [http://www.automation.siemens.com/wcmsnewscenter/details.aspx?xml=/content/10001666/en/gc/Pages/Adv141\\_S36\\_IndustrialWLAN.xml&xsl=publication-en-www4.xsl](http://www.automation.siemens.com/wcmsnewscenter/details.aspx?xml=/content/10001666/en/gc/Pages/Adv141_S36_IndustrialWLAN.xml&xsl=publication-en-www4.xsl).
- [6] M. Deruyck, E. Tanghe, W. Joseph, and L. Martens, "Characterization and optimization of the power consumption in wireless access networks by taking daily traffic variations into account," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, pp. 1-12, 2012.
- [7] I. Siomina, P. Värbrand, and D. Yuan, "Pilot power optimization and coverage control in WCDMA mobile networks," *Omega*, vol. 35, no. 6, pp. 683-696, 2007.
- [8] F. G. Debele, M. Meo, D. Renga, M. Ricca, and Y. Zhang, "Designing resource-on-demand strategies for dense WLANs," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2494-2509, 2015.
- [9] Y. Kim, G. Hwang, J. Um, S. Yoo, H. Jung, and S. Park, "Throughput performance optimization of super dense wireless networks with the renewal access protocol," *IEEE Transactions on Wireless Communications*, vol. 15, no. 5, pp. 3440-3452, 2016.
- [10] A. P. Jardosh, K. Papagiannaki, E. M. Belding, K. C. Almeroth, G. Iannaccone, and B. Vinnakota, "Green WLANs: on-demand WLAN infrastructures," *Mobile Networks and Applications*, vol. 14, no. 6, pp. 798-814, 2008.
- [11] M. A. Marsan, L. Chiaraviglio, D. Ciullo, and M. Meo, "A simple analytical model for the energy-efficient activation of access points in dense WLANs," in *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, Passau, Germany, 2010, pp. 159-168.
- [12] P. Bahl, M. T. Hajiaghay, K. Jain, S. V. Mirrokni, L. Qiu, and A. Saberi, "Cell breathing in wireless LANs: algorithms and evaluation," *IEEE Transactions on Mobile Computing*, vol. 6, no. 2, pp. 164-178, 2007.
- [13] W. Ikram, S. Petersen, P. Orten, and N. F. Thornhill, "Adaptive multi-channel transmission power control for industrial wireless instrumentation," *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 2, pp. 978-990, 2014.
- [14] Y. Yoon, and Y. H. Kim, "An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1473-1483, 2013.
- [15] X. Y. Zhang, J. Zhang, Y. J. Gong, Z. H. Zhan, W. N. Chen, and Y. Li, "Kuhn-munkres parallel genetic algorithm for the set cover problem and its application to large-scale wireless sensor networks," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 695-710, 2016.
- [16] V. P. Mhatre, K. Papagiannaki, and F. Baccelli, "Interference mitigation through power control in high density 802.11 WLANs," pp. 535-543.
- [17] A. M. Gibney, M. Klepal, and D. Pesch, "Agent-based optimization for large scale WLAN design," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 4, pp. 470-486, 2011.
- [18] J. Manicassamy, S. Sampath Kumar, M. Rangan, V. Ananth, T. Vengattaraman, and P. Dhavachelvan, "Gene suppressor: an added phase toward solving large scale optimization problems in genetic algorithm," *Applied Soft Computing*, vol. 35, pp. 214-226, 2015.
- [19] G. Sun, R. Zhao, and Y. Lan, "Joint operations algorithm for large-scale global optimization," *Applied Soft Computing*, vol. 38, pp. 1025-1039, 2016.
- [20] C. P. Chen, S. C. Mukhopadhyay, C. L. Chuang, T. S. Lin, M. S. Liao, Y. C. Wang, and J. A. Jiang, "A hybrid memetic framework for coverage optimization in wireless sensor networks," *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2309-2322, 2015.
- [21] X. Liu, "A deployment strategy for multiple types of requirements in wireless sensor networks," *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2364-2376, 2015.
- [22] N. Liu, D. Plets, K. Vanhecke, L. Martens, and W. Joseph, "Wireless indoor network planning for advanced exposure and installation cost minimization," *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 1, pp. 1-14, 2015.
- [23] M. Rebai, M. Le berre, H. Snoussi, F. Hnaïen, and L. Khokhi, "Sensor deployment optimization methods to achieve both coverage and connectivity in wireless sensor networks," *Computers & Operations Research*, vol. 59, pp. 11-21, 2015.
- [24] M. N. Omidvar, X. Li, and K. Tang, "Designing benchmark problems for large-scale continuous optimization," *Information Sciences*, vol. 316, pp. 419-436, 2015.
- [25] S. Mahdavi, M. E. Shiri, and S. Rahnamayan, "Metaheuristics in large-scale global continuous optimization: A survey," *Information Sciences*, vol. 295, pp. 407-428, 2015.
- [26] F. Mansourkiaie, L. S. Ismail, T. M. Elfouly, and M. H. Ahmed, "Maximizing Lifetime in Wireless Sensor Network for Structural Health Monitoring With and Without Energy Harvesting," *IEEE Access*, vol. 5, pp. 2383-2395, 2017.
- [27] Z. Y. Li, and M. S. Zhang, "Pin Assignment Optimization for Large-Scale High-Pin-Count BGA Packages Using Genetic Algorithm," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 5, no. 2, pp. 232-244, 2015.
- [28] D. Plets, E. Tanghe, A. Paepens, L. Martens, and W. Joseph, "WiFi network planning and intra-network interference issues in large industrial warehouses," pp. 1-5.
- [29] V. Angelakis, S. Papadakis, V. Siris, and A. Traganitis, "Adjacent channel interference in 802.11a: modeling and testbed validation," pp. 591-594.
- [30] L. Kong, M. Zhao, X. Y. Liu, J. Lu, Y. Liu, M. Y. Wu, and W. Shu, "Surface Coverage in Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 234-243, 2014.
- [31] X. Gong, T. De Pessemer, W. Joseph, and L. Martens, "A generic method for energy-efficient and energy-cost-effective production at the unit process level," *Journal of Cleaner Production*, vol. 113, pp. 508-522, 2016.
- [32] X. Gong, D. Plets, E. Tanghe, T. De Pessemer, L. Martens, and W. Joseph, "An efficient genetic algorithm for large-scale planning of dense and robust industrial wireless networks," *Expert Systems with Applications*, vol. 96, pp. 311-329, 2018.
- [33] W.-m. Hwu, "What is ahead for parallel computing," *Journal of Parallel and Distributed Computing*, vol. 74, no. 7, pp. 2574-2581, 2014.
- [34] X. Li, and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 2, pp. 210-224, 2012.
- [35] M. N. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 378-393, 2014.
- [36] Y. M. Wang, H. L. Yin, and J. Wang, "Genetic algorithm with new encoding scheme for job shop scheduling," *The International Journal of Advanced Manufacturing Technology*, vol. 44, no. 9, pp. 977-984, 2009.
- [37] P. Becue, B. Jooris, V. Sercu, S. Bouckaert, I. Moerman, and P. Demeester, "Remote control of robots for setting up mobility scenarios during wireless experiments in the IBBT w-iLab.t," *Testbeds and Research Infrastructure. Development of Networks and Communities: 8th International ICST Conference, TridentCom 2012, Thessaloniki, Greece, June 11-13, 2012, Revised Selected Papers*, T. Korakis, M. Zink and M. Ott, eds., pp. 425-426, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

- [38] D. Gorissen, I. Couckuyt, P. Demeester, T. Dhaene, and K. Crombecq, "A surrogate modeling and adaptive sampling toolbox for computer based design," *J. Mach. Learn. Res.*, vol. 11, pp. 2051-2055, 2010.
- [39] M. Hooshmand, S. M. R. Soroushmehr, P. Khadivi, S. Samavi, and S. Shirani, "Visual sensor network lifetime maximization by prioritized scheduling of nodes," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 409-419, 2013.
- [40] Y. Lin, J. Zhang, H. S. H. Chung, W. H. Ip, Y. Li, and Y. H. Shi, "An Ant Colony Optimization Approach for Maximizing the Lifetime of Heterogeneous Wireless Sensor Networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 3, pp. 408-420, 2012.
- [41] C. Zhao, C. Wu, J. Chai, X. Wang, X. Yang, J.-M. Lee, and M. J. Kim, "Decomposition-based multi-objective firefly algorithm for RFID network planning with uncertainty," *Applied Soft Computing*, vol. 55, pp. 549-564, 2017.